Multimodal Analysis for Object Classification and Event Detection

Multimodale analyse voor objectclassificatie en eventdetectie

Viktor Slavkovikj

Promotoren: prof. dr. S. Verstockt, prof. dr. ir. S. Van Hoecke
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Computerwetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. R. Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2015 - 2016

UNIVERSITEIT
GENT

ISBN 978-90-8578-905-5
NUR 980, 984
Wettelijk depot: D/2016/10.500/37

Acknowledgments

> *Today the robot is an accepted fact, but the principle has not been pushed far enough. In the twenty-first century the robot will take the place which slave labor occupied in ancient civilization. There is no reason at all why most of this should not come to pass in less than a century, freeing mankind to pursue its higher aspirations.*
>
> -Nikola Tesla-

Almost four-and-a-half years ago, I embarked on a journey which has finally lead to the completion of this dissertation. Reflecting now on my time as a PhD student, I can say that it was an interesting and challenging trip, and, overall, a very rewarding experience. Many people selflessly contributed along the way, without whom this work would not have been the same.

First of all, I would like to thank prof. Rik Van de Walle, for giving me the opportunity to start a PhD at Multimedia Lab (now Data Science Lab) at Ghent University. During my work in the lab, I have always had the liberty to explore new research ideas, and the chance to build my professional skills. Rik, thank you for all your support.

I am also very grateful to have prof. Steven Verstockt and prof. Sofie Van Hoecke as my advisers. Steven and Sofie guided me throughout my PhD and were always available, whether it was for questions, technical discussions, or reviewing papers. Their comments and suggestions were indispensable to improving my research work. Sofie and Steven, a big thank you to both of you.

A word of thanks also goes to the members of the examination board. Your reading reports and comments greatly improved the quality of this work.

I would also like to give credit to the current and former colleagues of the lab. It is your enthusiasm and companionship that contributed to an enjoyable ambience at work. I also thank Ellen Lammens, Laura Smekens, and Kristof Van Damme for their administrative and logistic support.

A special thanks goes to my closest colleagues and office mates: Baptist, Olivier, Fréderic, Azarakhsh, Abhineshwar, Florian, Mi Jung, and Jasper. I feel privileged for the time spent in your company. You were always ready to help and to offer your insights into problems. All the laughs we shared fostered a relaxed and pleasant working environment.

A special thanks also goes to prof. Wesley De Neve. Wesley, thank you for organizing our SaVI team, and for putting in extra hours of work whenever any of the SaVI-ers had to meet an important conference deadline.

Finally, words are not enough to express how grateful I am for all the love and support from my family: my father Zoran, my mother Javorka, and my sister Marija. Ви благодарам за безрезервната поддршка, затоа што секогаш сте покрај мене, за целата Ваша љубов...

Viktor Slavkovikj
Ghent, June 2016

Summary

Object classification and event detection are two principal problems in computer vision and artificial intelligence. Being able to correctly identify the types of objects in the environment, and to detect the occurrence of events, are prerequisites in constructing machines capable of interacting with the environment and performing intelligent tasks. Therefore, solving the different instances of classification and detection problems extends the useful capabilities of such machines. By making use of different information modalities, in this dissertation, we will investigate instances of such problems, specifically, classification of terrains, detection of component faults in rotating machines, and detection of large-scale events such as wildfires.

Automatic content based classification of terrains finds many applications. Whether it is simply choosing a route suggestion for a recreational bicycle ride, or navigation of an autonomous robot, knowledge over the type of road or terrain traversed can have an important influence. In recent years, with the advance of remote-sensing imaging sensors, it has also become possible to analyze the terrain surface of large areas. Accurate classification of vast terrains is essential for several fields. For example, it can be used for agricultural purposes in order to monitor the condition of different crops, or in the context of environmental science for monitoring the level of ocean pollution. It is also useful in the detection of potential areas containing mineral deposits in geology and mineralogy, and for different surveillance scenarios. Additionally, the same imaging technology used for remote-sensing applications is also used in areas such as forensics, biomedicine, food sciences, and information security.

We will start our investigation with road and terrain type classification in Chapter 2, where a multimodal road classification system is proposed. The goal of the proposed system is to differentiate between

several classes of roads, which can be used for automatic annotation of routes. A bicycle sensing setup will be employed to collect visual and vibration data modalities along a route, which will then be used in identifying different road categories. We will show how, through training a nonlinear classifier on features engineered from the road data modalities, a good classification accuracy can be achieved on real-world data.

Additionally, we will investigate how available online images can be used for road classification. For this purpose, a comprehensive road image dataset is constructed from user contributed route information and images queried from available online services. We will show how, by learning features from online road images, high classification results can be obtained for paved and unpaved road categories that are comparable to those obtained by a fine tuned feature engineered system.

Large-scale terrain analysis, through classification of hyperspectral remote-sensing images, is investigated in Chapters 3 and 4, where two general methods for unsupervised and supervised hyperspectral image classification are proposed. For hyperspectral image data, it is challenging to derive relevant features, which is essential for obtaining good results in a feature engineered classification framework. In Chapter 3, we will, therefore, investigate how spectral information can be more effectively utilized through feature learning from subsets of the correlated hyperspectral bands. We will show how, by unsupervised learning of basis functions, and by nonlinear encoding of spectral input samples, discrimination between different terrain classes can be achieved.

Chapter 4 builds on the observations made in Chapter 3, and further investigates the problem of hyperspectral image classification, but from a perspective of supervised learning, and under a low number of training samples. By constructing a layered neural network model, we will present an end-to-end system for hyperspectral image classification. Our experimental results show that this integral classification method is able to learn structured feature representations without incorporating prior knowledge for the problem. Furthermore, the features learned resemble different spectral band-pass filters, which gives an interesting and novel insight to hyperspectral image classification.

The relevance of the method proposed in Chapter 4 is not only pertinent to the problem of hyperspectral image classification. We will

show, in Chapter 5, how a model similar to the one initially developed for hyperspectral image classification can be used in the classification of different faults that can occur in rotating machines. By learning the joint spatial patterns from vibration signals of machines such as wind turbines, our method can also be used for condition monitoring, and for prolonging the operational uptimes of the machines.

Accurate classification of vast terrains is directly related to the problem of management of large-scale events such as wildfires. For example, having information over the type of vegetation in a large area can significantly aid in the prevention of fire hazards, and contribute in choosing an appropriate response in case of a wildfire. Complementary to this kind of knowledge, in Appendix A, we will investigate the potential of social media information for detection and management of wildfires. Current mobile communication devices are widely available, and contain multiple sensors, which has revolutionized the generation and exchange of user information. The ability to effortlessly produce different sensory output has transformed the role of the users, from passive consumers of news, to active contributors of quantifiable information. The popularity of social media platforms has led to the creation of active user communities, which can be considered as a kind of human-centric sensor networks. The combined user-generated output of these networks can provide a very valuable source of information. Compared to traditional information sources, human-centric sensing can offer several advantages such as immediacy, and specificity of shared information. Through a review of systems, methods, and applications, which utilize different social media information modalities, we will investigate the possibilities of collaborative human-centric sensing for early detection and management of large-scale disasters. Additionally, based on the reviewed systems and methods, we will delineate the major components that would be necessary in future social media systems for wildfire detection and management.

Finally, we hope to convince the reader of the relevance of the work presented in this dissertation. The long term goal of this research work is to contribute to the joint effort of building useful and intelligent systems that can be employed for a wide range of current as well as future practical applications.

Samenvatting

Objecten classificeren en events detecteren zijn twee fundamentele problemen in computer visie en artificiële intelligentie. De types van objecten correct identificeren in verschillende omgevingen, en het detecteren van events zijn vereisten om machines te creëren die met de omgeving kunnen interageren en om machines intelligente taken uit te laten voeren. Door de verschillende classificatie en detectie problemen op te lossen worden de bekwaamheden van dergelijke machines uitgebreid. Gebruikmakend van verschillende types informatie zullen in deze thesis drie voorbeelden van zulke problemen opgelost worden. Meer specifiek zal er gefocust worden op het classificeren van terreinen, i.e., ondergronden, het detecteren van component fouten in roterende machines, en het detecteren van grootschalige events zoals natuurbranden.

Automatische classificatie van terreinen is terug te vinden in veel applicaties. Of het nu simpelweg het aanbevelen van routes is voor recreatief fietsen, of een robot autonoom laten navigeren, de kennis over het type wegdek of terrein zal een belangrijke rol spelen. Recentelijk, met de opkomst van remote-sensing imaging sensoren, is het ook mogelijk geworden om het terrein van grote oppervlakten te analyseren, bijvoorbeeld met behulp van drones. Het accuraat classificeren van grote terreinen is essentieel voor verschillende sectoren. In de landbouw, bijvoorbeeld, kan het gebruikt worden om de staat te monitoren van verschillende gewassen; in de milieukunde kan het gebruikt worden om de vervuiling van oceanen te monitoren. Het is ook mogelijk om hiermee gebieden te detecteren die potentieel mineralen bevatten, alsook kan het gebruikt worden in de context van toezicht scenario's. Dezelfde technologie kan ook gebruikt worden voor remote-sensing applicaties voor forensisch onderzoek, medische biologie, voedingswetenschappen en informatiebeveiliging.

We beginnen ons onderzoek met wegdektype en terreintype classificatie in hoofdstuk 2, waarin een multimodaal classificatiesysteem wordt voorgesteld. Het doel van het systeem is om een onderscheid te kunnen maken tussen verschillende types van wegdek zodat deze gebruikt kunnen worden voor automatische routeannotatie. Een fiets gebaseerd systeem wordt gebruikt om visuele en vibratie data te verzamelen gedurende een fietsrit. Hierna wordt de data gebruikt voor om verschillende wegdekcategorieën te identificeren. We tonen aan dat door een non-lineair classificatiemodel te trainen gebruikmakende van handgemaakte features, een goede accuraatheid behaald kan worden op echte data.

We onderzoeken tevens ook hoe online afbeeldingen gebruikt kunnen worden voor wegdekclassificatie. Hiervoor is een grote dataset gecreëerd bestaande uit afbeeldingen van wegdekken. Deze is gemaakt door gebruikers en door online services te bevragen. We tonen aan dat, door features te leren op basis van online wegdek afbeeldingen, een hoge accuraatheid kan behaald worden voor geplaveide en niet-geplaveide wegdekken. Deze resultaten zijn vergelijkbaar met de resultaten behaald op basis van de fietssensoren.

Grootschalige terreinanalyse aan de hand van hyperspectrale afbeeldingen wordt onderzocht in hoofdstukken 3 en 4. In deze hoofdstukken worden twee algemene methoden voorgesteld om hyperspectrale afbeeldingen te classificeren. De eerste methode is ongesuperviseerd en de tweede methode is gesuperviseerd. Om goede classificatieresultaten te behalen, wanneer hyperspectrale data wordt gebruikt, moeten relevante features gecreëerd worden. In hoofdstuk 3, onderzoeken we hoe spectrale informatie meer effectief gebruikt kan worden door features te leren op basis van subsets van gecorreleerde hyperspectrale banden. We tonen aan dat door ongesuperviseerde basisfuncties te leren en non-lineaire codering van spectrale input samples te gebruiken, onderscheiding tussen verschillende terreinklassen uitgevoerd kan worden.

Hoofdstuk 4 bouwt voort op de observaties gemaakt in hoofdstuk 3, en verricht ook onderzoek met betrekking tot classificatie van hyperspectrale beelden. Deze keer wordt dit gedaan aan de hand van gesuperviseerd leren van features op basis van een kleine set van training samples. We presenteren een neuraal netwerk model dat een end-to-end hyperspectraal afbeelding classificatiesysteem is. Onze experimenten tonen aan dat deze classificatiemethode gestructureerde

features kan leren zonder de vereiste om voorafgaande kennis te incorporeren in het systeem. Deze geleerde features zijn gelijkaardig met verschillende spectrale band-pass filters. De features geven nieuwe en interessante inzichten in de classificatie van hyperspectrale beelden. De methode voorgesteld in hoofdstuk 4 is niet alleen toepasbaar op het hyperspectraal classificatie probleem. We tonen ook aan, in hoofdstuk 5, hoe een gelijkaardig model gebruikt kan worden voor het classificeren van verschillende fouten in roterende machines. Door het leren van spatiale patronen van vibratiedata van machines zoals wind turbines, kan onze methode ook gebruikt worden voor het monitoren van de conditie van deze machines zodat de operationele uptime verbeterd kan worden.

Het accuraat classificeren van grote terreinoppervlakten is direct gerelateerd met het probleem van grootschalige events te managen zoals natuurbranden. De informatie over de vegetatietypes, bijvoorbeeld, in een groot gebied kan het brandgevaar helpen reduceren en helpen bij de keuze van de gepaste reactie in het geval van brand. Complementair met dit type kennis is rampgebaseerde informatie, die steeds meer en meer beschikbaar komt via sociale media. De dag van vandaag zijn mobiele communicatie toestellen overal beschikbaar. Ze bevatten verschillende sensoren die tot een revolutie geleid hebben in de generatie en uitwisseling van gebruikersinformatie. De mogelijkheid om data via verschillende sensoren zonder moeite te genereren heeft de rol van de gebruiker veranderd van passieve nieuwslezer tot actieve bijdrager van informatie. De populariteit van sociale media platformen heeft geleid tot de creatie van actieve gebruikersgroepen die kunnen beschouwd worden als een vorm van menselijke sensornetwerken. De gebruiker gegenereerde output van deze netwerken kan een belangrijke bron van informatie zijn. In vergelijking met traditionele informatiebronnen biedt human-centric sensing grote voordelen, zoals de onmiddellijke beschikbaarheid en specificiteit van gedeelde informatie. Aan de hand van een review van systemen, methoden en applicaties die verschillende sociale media informatie modaliteiten gebruiken, onderzoeken we de mogelijke samenwerkingen tussen human-centric sensing voor vroegtijdige detectie en management van grootschalige rampen. Bijkomend, op basis van onze review, lijsten we de vereiste componenten op die nodig zijn voor toekomstige sociale media gebaseerde systemen om bij te dragen in het detecteren en managen van natuurbranden.

We hopen de lezer te overtuigen van de belangrijkheid van het werk in deze thesis. Ons langetermijndoel van dit onderzoek is om bij te dragen tot een gezamenlijke poging om nuttige en intelligente systemen te maken die kunnen worden gebruikt voor een breed gamma van actuele en toekomstige praktische applicaties.

List of abbreviations

AE	autoencoder.
ANN	artificial neural network.
API	application programming interface.
BPFO	ball pass frequency of the outer raceway.
CM	condition monitoring.
CNN	convolutional neural network.
DBN	deep belief network.
DFT	discrete Fourier transform.
DWT	discrete wavelet transform.
EILB	extremely inadequately lubricated bearing.
EILB-IM	extremely inadequately lubricated bearing during imbalance.
EM	expectation-maximization.
GPS	global positioning system.
GPU	graphics processing unit.
HB	healthy bearing.
HB-IM	healthy bearing during imbalance.
HMM	hidden Markov model.
HSI	hyperspectral image.
ICA	independent component analysis.
ICT	information and communication technology.
LDA	linear discriminant analysis.
LE	Laplacian eigenmaps.
LLE	locally linear embedding.
LLTSA	linear local tangent space alignment.
LPP	locality preserving projection.

LTSA	local tangent space alignment.
MILB	mildly inadequately lubricated bearing.
MILB-IM	mildly inadequately lubricated bearing during imbalance.
NN	neural network.
NPE	neighborhood preserving embedding.
NWFE	nonparametric weighted feature extraction.
ORF	outer-raceway fault.
ORF-IM	outer-raceway fault during imbalance.
P2P	peer-to-peer.
PCA	principal component analysis.
POS	part-of-speech.
RBF	radial basis function.
RBM	restricted Boltzmann machine.
RF	random forest.
RGB	red, green, and blue image channels.
RMS	root-mean-square.
RSS	really simple syndication.
SDA	semi-supervised discriminant analysis.
SELD	semi-supervised local discriminant analysis.
SELF	semi-supervised local Fisher discriminant analysis.
SPP	sparsity preserving projections.
SURF	speeded up robust features.
SVM	support vector machine.

Contents

Chapter 1

Introduction

The main focus of this dissertation is on problems of object classification and event detection through the use of different information modalities. In this chapter, we describe the broader context and delineate the specific tasks addressed by our research. Additionally, we give an introduction to the particular data modalities used throughout this dissertation, and the general machine learning approaches which will figure in our proposed methods. We conclude the chapter with an outline of our research, and an overview of published manuscripts.

1.1 Context

Higher level cognitive tasks such as object classification and event detection are two challenging computer vision and artificial intelligence problems. In computer vision, object classification is the problem of determining the specific class (such as person, boat, bird) an object belongs to from digital images containing instances of predefined classes or categories (Figure 1.1). On the other hand, event detection can be formulated as a binary classification problem and deals in general with detection of a specific irregularity or anomaly occuring in the data.

In this dissertation, we focus on specific instances of the object classification and event detection problems: terrain or road type classification, detection of faults in rotating machines, and wildfire event detection. In the terrain classification task, our goal is to utilize sensory data to automatically determine types of roads or terrains, such

as the road surfaces found along a cycling route (Figure 2.5), or the different land-cover classes from a high-altitude scan of an area (Figure 3.3). Our objective for the task of fault detection in rotating machines is to detect events that deviate from the normal operating modes of the machines. Timely detection of faulty components facilitates condition monitoring of machines, which is necessary for enhancing their correct operation. On the other hand, in the case of wildfire event detection, we are interested in the use of collective intelligence data, which can be beneficial for detection and management of large-scale disaster events, and we review methods and systems that can be utilized in the case of wildfire events.

(a) (b) (c)

Figure 1.1: Example instances from the person (a), boat (b), and bird (c) classes of the Pascal VOC [1] dataset.

Object classification is a fundamental problem in computer vision, and has drawn the interest of researchers for several decades. The challenges in generic object classification arise from several intrinsic variabilities of the different class instances. Among the most important of these variabilities are the high intra-class and low inter-class differences in the categories of interest. Both impose a robustness requirement on a classifier. The former requirement is in terms of the ability of the classifier to categorize different instances of the same class, where the objects of the class can differ in a number of attributes, such as appearance, pose, size, scale, shape, position, viewpoint, and context. The latter refers to the ability of the classifier to discriminate between objects which are characterized by some

similar attributes, but which nevertheless belong to different classes. For example, in Figure 1.1 (a), there are actually two objects which can be categorized as persons (a man and a small child), which have different visual appearances. Similarly, Figure 1.1 (b) contains one specific instance of a broad class of boats. Contrary to this, if we try to categorize the specific type of bird in Figure 1.1 (c), we would likely find it difficult to disambiguate between several classes of small birds due to inter-class similarities. Furthermore, besides challenges in terms of robustness, there are also scalability challenges such as the dimensionality of the input representation of object instances, and the number of object categories.

Given the complexity of the object classification problem, research efforts have been focused on contriving feature representations which would allow sufficient discrimination between instances of different classes while being invariant to the many within-class variations. Another way in which researchers have approached object classification problems is by exploiting task-complementary multimodal information.

1.2 Multimodal information

Multimodal information is information that originates from multiple sources or modalities. For example, two complementary information modalities for understanding speech are audio and video. By using both acoustic and visible information (the movement of the speaker's lips), the intelligibility of speech can be improved [2].

Although information modalities are typically related to specific physical detectors used to register the information, such as a camera and a microphone in the case of multimodal speech recognition, we are essentially interested in whether a modality offers a different kind of information for the problem at hand. Therefore, it makes sense to consider a modality as an abstraction of a detector, and to relate it to a probability distribution from which the measured quantities are being generated.

Appart from additional complementary information offered by different modalities, in cases when information from one modality is missing or corrupted, data from existing modalities can be used to replace or infer the missing information. In this dissertation, we also adopt a multimodal information approach for the tasks of terrain clas-

sification and wildfire event detection, by relying on several modalities such as visual and vibration information, or visual and infrared information for terrain classification, and different types of visual, textual, and geographic information for the case of wildfire detection. Hereinafter, we will describe two sources of multimodal data that are used in this thesis: hyperspectral imagery and social media.

1.2.1 Hyperspectral imaging

In recent years, developments of electro-optical technology have led to the emergence of high-resolution imaging spectroscopy sensors. Compared to regular digital color cameras, which capture only three image channels from the red, green, and blue region of the visible spectrum, imaging spectrometers can acquire measurements from continuous narrow bands of a wide interval of the electromagnetic spectrum. The resulting hyperspectral images (Figure 1.2) consist of hundreds of image channels, typically in the wavelength interval of 0.4–2.5 μm of the electromagnetic spectrum. Spanning both the visual and the

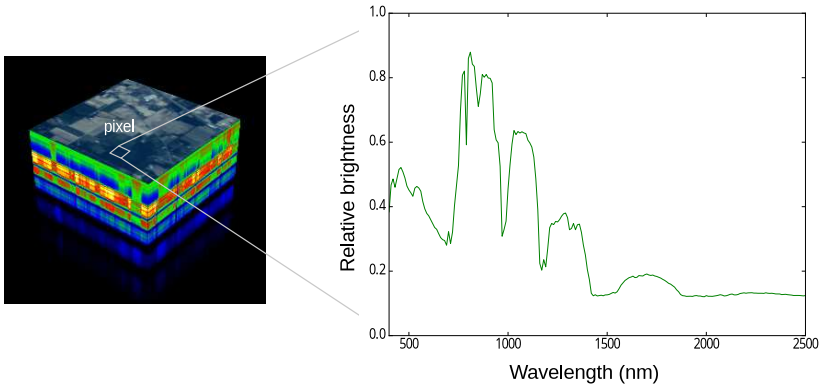


Figure 1.2: A visualization of a hyperspectral image together with a plot of the spectrum of a single hyperspectral pixel.

reflected infrared parts of light, hyperspectral imaging sensors act as sources of multimodal (bi-modal) information. Because of the large quantities of highly discriminative multimodal information, coupled with the possibility of high-resolution acquisition, hyperspectral imagery offers ample opportunity for analysis in several fields. As such, analysis of hyperspectral images finds application in areas such as

geology and mineralogy, agriculture, food sciences, astronomy, environmental science, surveillance, disaster management, and forensics.

Hyperspectral image classification is a central component in the analysis of hyperspectral images, and we will also address hyperspectral image classification in this thesis in the context of terrain classification of remote-sensing scenes—a reoccuring problem in many of the aforementioned application areas. This, however, has proven to be a challenging task notwithstanding the rich hyperspectral information, since terrain classification from remote-sensing hyperspectral images is plagued with similar difficulties as generic object classification discussed in the beginning. In particular, hyperspectral remote-sensing images are acquired by mounting an imaging spectroscope on an aircraft or satellite, which in turn is used to scan the area of interest from high altitude and detect light reflected from the scanned surface (Figure 1.3).

There are several factors which can cause an increase of intra-class and a decrease of inter-class variations in hyperspectral remote-sensing images. The inherent variability in the structure and composition of materials of the same class results in differences of the materials' spectral signatures. Furthermore, due to the specific process of acquisition, both intra-class and inter-class variations can be induced as a result of atmospheric effects such as scattering and absorption of light [3]. Additionally, intra-class and inter-class variations occur due to the topology of the terrain, because the reflected light depends on the impact angle of light incident to the surface normal [4]. Also, shadows cast from tree canopy, buildings, or clouds can significantly change the spectral characteristics of objects. Finally, light reflected from adjacent terrains belonging to different terrain categories produces mixed spectral measurements, thus rendering classification of such terrains difficult.

No less challenging than the described intra-class and inter-class variations are the scalability requirements for a hyperspectral image classifier. Namely, the highly discriminative hyperspectral data come with the curse of dimensionality. The curse of dimensionality is a term used to state that the number of samples necessary to uniformly cover a region of space rises exponentially as the number of dimensions of the space increase. Therefore, for a fixed number of samples, the predictive power of a classifier initially grows to an optimum as the dimensionality of the samples increases, and then diminishes with

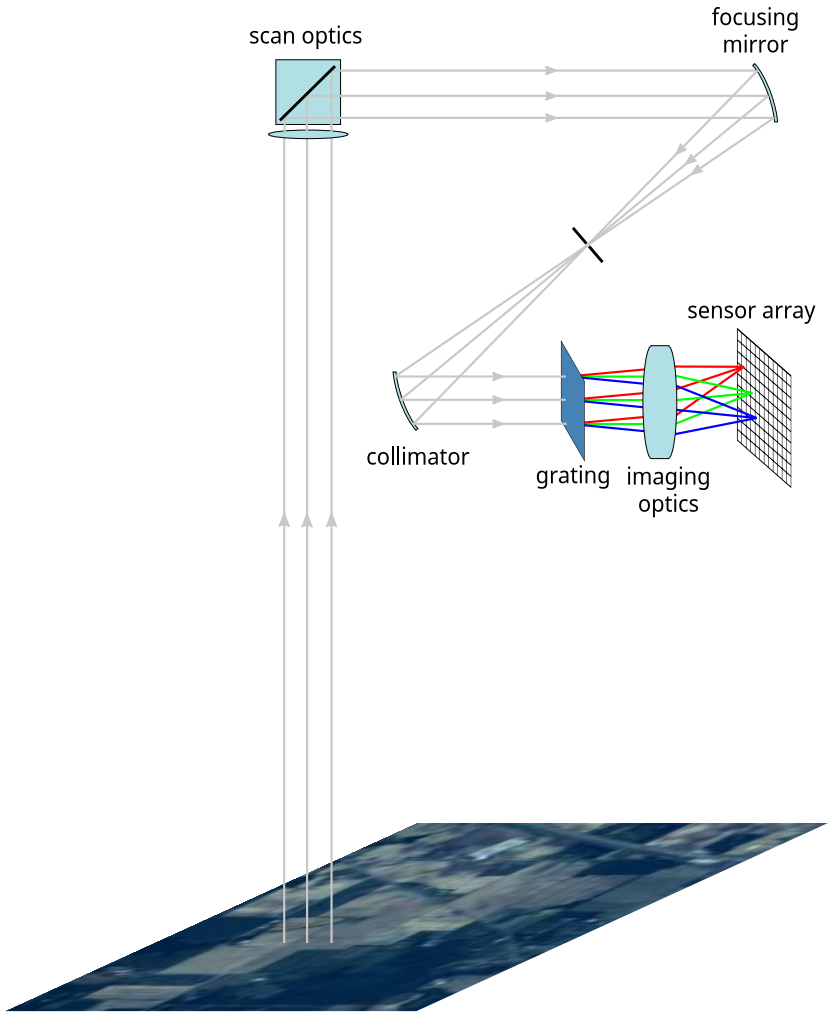


Figure 1.3: A diagram of a hyperspectral imaging sensor used in remote-sensing.

further increase of the dimensionality [5].

Despite the challenges associated with hyperspectral images, terrain classification of hyperspectral imagery of remote-sensed scenes remains a cost-efficient method for analyzing the surface of large areas. With recent commercialization of imaging spectroscopy, hyperspectral imaging data has also been made available to the broader public. This will likely lead to a further increase in the application of hyperspectral information for scientific and civilian purposes.

1.2.2 Social media information

Unlike hyperspectral imaging, where information is gathered by using a specialized sensor in a centralized manner, distributed data sensing, through monitoring and analysis of content generated by a large group of individuals, can also serve as a valuable source of information. Together with the introduction of smart mobile devices, an implicit proliferation of inexpensive sensors has also taken place. Today, there are over two billion smartphone devices in use worldwide, and it is estimated that this number will raise to six billion by 2020[1]. Cur-

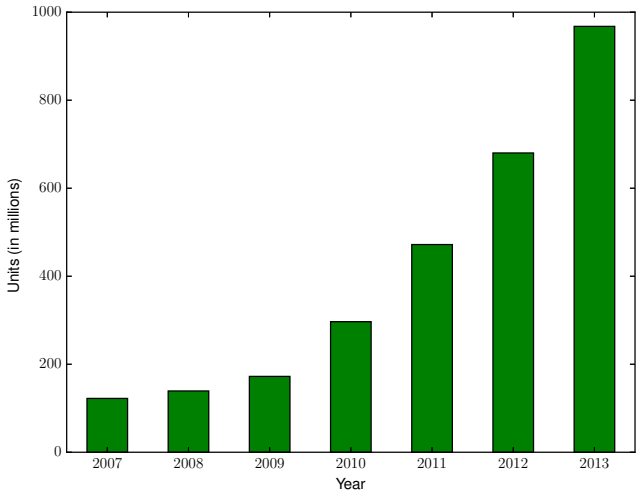


Figure 1.4: Number of smartphones sold worldwide from 2007 to 2013.[2]

rent smartphones are commonly equipped with several sensors, such

[1] http://goo.gl/KJ29mV
[2] http://goo.gl/h9e5WL

as a camera, accelerometer, and a global positioning system. The large quantities of these multisensor mobile devices have had a transformative effect on the role of the user—from mainly a consumer of information, to a potential provider of quantifiable multimodal sensory data, and a part of a larger-scale participatory sensing system. Furthermore, the emergence of social networking platforms such as Twitter, Facebook, Flickr, and YouTube have revolutionized the way of communication between individuals and groups by simplifying information sharing between users. The timely and brief character of information created by millions of social media users allows for a globally distributed information and news sharing network, which often surpasses that of traditional media outlets in terms of immediacy and value of the delivered information, especially in cases of large-scale events.

There are, however, challenges when using social media information for automatic event detection tasks. The huge volume of user-generated information requires efficient and scalable methods to be able to process the data. Additionally, the information stream contains a lot of uninformative posts for the tasks at hand, or misleading information. Social media messages are also short and do not provide much context, can lack structure, contain mixed language, abbreviations, and syntax errors [6]. In the context of this thesis, we will rely on different social media information modalities, both in Chapter 2, where we use collaborative information sensing for automatic terrain annotation of routes, and again in Appendix A, in the context of our review of wildfire risk management from social media information.

1.3 Supervised learning and unsupervised feature learning

For many complex problems in computer vision and artificial intelligence, such as classification and event detection, it is difficult to reach satisfying solutions by programming machines using explicit rules. This is especially true when using multimodal information, where some of the modalities cannot be easily visualized and analyzed by humans experts. Instead, a central goal of machine learning is to develop algorithms that rely on data to "teach" machines how to solve problems by example—similar to how humans learn to tackle problems from experience.

Let $\mathcal{D}$ denote a set of training examples $\{(\mathbf{x}_n, \mathbf{t}_n)\}, n = 1, \ldots, N$, such that each example is a pair comprising of an input vector $\mathbf{x}$ and its corresponding target $\mathbf{t}$. For example, given a hyperspectral pixel of a remote-sensing scene as input, its corresponding target could be the discrete terrain class label assigned to this pixel. Given such a dataset $\mathcal{D}$, the goal of supervised learning is to train a function $y(\mathbf{x}, \theta)$ capable of correctly approximating or predicting the value of the target $\mathbf{t}$ for a novel input $\mathbf{x}$. The function $y(\mathbf{x}, \theta)$ is constructed by optimizing a set of adjustable parameters θ. The process of optimization, or training, is governed by a loss function $\mathcal{L}(\mathcal{D}, \theta)$, which is used to quantify the discrepancy between the value predicted by our function y and the desired output $\mathbf{t}$ over the examples in the dataset $\mathcal{D}$. Therefore, the objective of training is to minimize the overall loss, or to maximize the negative of the loss.

Due to the complex nonlinear relationship between the raw inputs and their targets, learning a predictive function y directly from input can prove to be an unattainable goal for any nontrivial problem, even when using data only from a single modality. Learning such a function or model may require optimizing a huge number of adaptive parameters, which in the case of a limited number of labeled training examples would likely lead to overfitting. Overfitting occurs when the learned model produces very good predictions for the examples that it has already seen during training, but gives poor results for new data from the same distribution. To facilitate learning, researchers have designed alternative representations of the input in terms of features. Features can be seen as attributes which provide a succinct description of the most important characteristics of the data. For example, in visual images, the contour and color of an object can be important attributes for classification.

Although feature engineering produces good results, designing relevant features is very task specific. Furthermore, it can be exceptionally difficult to come up with robust features for complex problems, and for data modalities for which there is no clear understanding of the characteristics that may produce good results. Recently, however, there have been significant research efforts to develop methods which can learn feature representations from data in an unsupervised manner. The goal of such unsupervised feature learning algorithms is to learn a new, more expressive representation of the input by employing input data without accompanying target values, which is easier

to obtain compared to labeled data. The newly learned feature representations can then be used to alleviate the (possibly supervised) learning of higher level tasks. We will explore both supervised learning and unsupervised feature learning approaches in solving problems addressed in this dissertation.

1.4 Outline

The rest of this dissertation is organized as follows. In Chapter 2, a multimodal system for terrain classification is proposed. The system employs visual and vibration data to discriminate between different terrain or road categories, which can be useful for automatic route annotation and autonomous robot navigation. Additionally, we will show how we can distinguish between paved and unpaved road classes with high intra-class variances by using unsupervised feature learning to build a model from online images.

Chapters 3 and 4 deal with classification of hyperspectral images. In Chapter 3, we will see how we can exploit spectral information by learning sub-feature basis representations in the spectral domain. In Chapter 4, we build on the insights gained in Chapter 3 and propose a novel integral approach for hyperspectral image classification based on convolutional neural networks.

The generality of the method developed in Chapter 4 is further explored in the context of condition monitoring of machines. Therefore, in Chapter 5, we will show how a similar model trained on vibration data can be used to discriminate between different faults occuring in rotating machines. We refer the reader to the work of Janssens et al. [7, 8] for more information on fault detection and condition monitoring of machines.

In Appendix A, we will investigate the detection and management of wildfire events—an area where large-scale terrain classification can play a significant role in prevention, monitoring, and impact estimation. Here, however, we will focus on using social media information, and provide a review of methods, applications, and systems which can use such social media modalities in the detection of wildfires.

The use of social media information is complementary to fire analysis methods based on traditional sensors such as visual and thermal cameras. In the latter context, several methods and systems[3] for in-

[3] `https://goo.gl/DrS2yY`

door and outdoor fire analysis have been investigated in the work of Verstockt et al. [9–11]. Additionally, in the domain of social media, the works of Godin et al. [6,12,13] and Vandersmissen et al. [14] deal with the analysis of short messages and videos, which is important for inference of information from this domain.

Finally, this dissertation ends with Chapter 6, where we summarize the most important observations and major conclusions that can be drawn from our research.

1.5 Publications overview

The research activities leading to this dissertation have been published in several international peer-reviewed journals and conferences:

1.5.1 A1 publications

1. Viktor Slavkovikj, Steven Verstockt, Sofie Van Hoecke, and Rik Van de Walle. Review of wildfire detection using social media. *Fire Safety Journal, vol. 68, no. 0, pp. 109–118, 2014.*

2. Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Unsupervised spectral sub-feature learning for hyperspectral image classification. *International Journal of Remote Sensing vol. 37, no. 2, pp. 309–326, 2016.*

3. Olivier Janssens, Raiko Schulz, Viktor Slavkovikj, Kurt Stockman, Mia Loccufier, Rik Van de Walle, and Sofie Van Hoecke. Thermal image based fault diagnosis for rotating machinery. *Infrared Physics & Technology, vol. 73, pp. 78–87, 2015.*

4. Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, and Rik Van de Walle. Collaborative bike sensing for automatic geographic enrichment. *IEEE Signal Processing Magazine, vol. 31, no. 5, pp. 101–111, 2014.*

5. Olivier Janssens[4], Viktor Slavkovikj[4], Bram Vervisch, Steven Verstockt, Kurt Stockman, Mia Loccufier, Rik Van de Walle, and Sofie Van Hoecke. Convolutional Neural Network Based

[4]Contributed equally to this work.

Fault Detection for Rotating Machinery. *Journal of Sound and Vibration (accepted for publication).*

1.5.2 C1/P1 publications

1. Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Image-based road type classification. *Pattern Recognition (ICPR), 22nd International Conference on, Aug 2014, pp. 2359–2364.*

2. Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Hyperspectral image classification with convolutional neural networks. *ACM MM'15.*

3. Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Visitor-art interaction by motion path detection. *18th International Conference on Digital Signal Processing (DSP), 2013.*

4. Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, Jürgen Slowack, and Rik Van de Walle. Multi-modal bike sensing for automatic geo-annotation: geo-annotation of road/terrain type by participatory bike-sensing. *10th International Conference on Signal Processing and Multimedia Applications, Proceedings, 2013, pp. 39–49.*

5. Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for Twitter hashtag recommendation. *22nd International Conference on World Wide Web companion, Proceedings, 2013, pp. 593–596.*

6. Steven Verstockt, Viktor Slavkovikj, Olivier Janssens, Pieterjan De Potter, Jürgen Slowack, and Rik Van de Walle. Web-based enrichment of bike sensor data for automatic geo-annotation. *GEOCROWD 2013, Proceedings.*

7. Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, Baptist Vandersmissen, Jürgen Slowack, and Rik Van de Walle. Automatic geo-mashup generation of outdoor activities. *International Conference on Advances in Mobile Computing & Multimedia, Proceedings, 2013.*

8. Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, Olivier Janssens, Jürgen Slowack, and Rik Van de Walle. Automatic geographic enrichment by multi-modal bike sensing. *Communications in Computer and Information Science 456, 2014, pp. 369–384.*

9. Steven Verstockt, Viktor Slavkovikj, and Kevin Baker. Map-based linking of geographic user and content profiles for hyper-local content recommendation. *17th International Conference, HCI International 2015, Proceedings. pp. 53–63*

Chapter 2

Terrain Classification

The ability to automatically determine the road type from sensor data is of great significance for automatic annotation of routes and autonomous navigation of robots and vehicles. In this chapter, we present a road/terrain classification system based on multimodal data analysis. The system makes use of accelerometer and visual sensor information, obtained from participatory recreational cyclists or online mapping platforms, to determine the terrain type. Our proposed approach relies on common hardware, and can make use of available online information to categorize roads. Experimental results on challenging real-world data show that the proposed system can achieve high accuracy in road/terrain classification.

2.1 Introduction

The advance of sensor technology, coupled with increasing on-board processing capabilities of current smartphone devices, has enabled users to efficiently create, capture, and share information about their activities. Furthermore, the availability of different sensors on modern smartphones allows users to act as sensor operators, i.e., to contribute sensor measurements about their environment. At the same time, the aggregation of these sensory information from many participants can contribute to a larger-scale effort to create common knowledge about a particular area of interest. In the field of geographic information systems, for example, the tendency of participatory data collection has had a propitious impact. Where the process of mapping the Earth has

been the task of a small group of people (surveyors, cartographers, and geographers) for many years, recently, it has become possible for everyone to participate in several types of collaborative geographic projects. The abundance of user-generated sensor information has resulted in the creation of web-based systems which provide different services from analyses of the aggregated user data. Online geographic information systems such as OpenStreetMap[1], RouteYou [15], and Bikemap[2] rely heavily on user-contributed sensor data to offer location oriented services.

Two common goals of this kind of systems are to provide querying of locations on interactive maps, and discovery of routes for recreational GPS-users such as cyclists and hikers. The latter makes use of pre-created GPS trajectories submitted by the users, while the former utilizes user annotations of objects and infrastructure. For route finding, it has been shown [16] that the road type or terrain characteristics have an important influence on route ranking. Therefore, it is not surprising that people try to annotate the type of the route they are submitting to allow for an effective search of good routes for fellow users. As opposed to route recording, annotating the different parts of a route requires active user involvement, and is both laborious and error prone. In this chapter, we investigate the possibility of

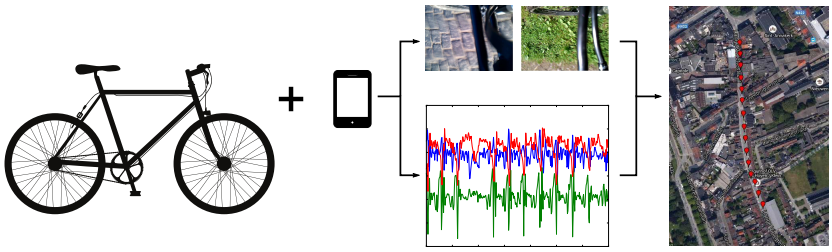


Figure 2.1: General overview of the bicycle sensing setup.

automatic classification and annotation of road or terrain type from mobile sensor data (e.g., obtained from a smartphone mounted on a bicycle), or from road images available on the web. Specifically, the system proposed here relies on image and accelerometer data captured by the users' smartphones. A general overview of the proposed setup is shown in Figure 2.1. The smartphone and GPS data are

[1] http://www.openstreetmap.org
[2] http://www.bikemap.net/en/

collected using the on-board device(s) mounted at the bike's handlebar. For the collection of the accelerometer data, the device can be placed or stored to the user's preference. For the camera sensor, the terrain/road needs to be in the field-of-view of the camera. However, in this chapter, we will also present a separate method for content-based road type classification from images only, which we evaluate on available online images from geographic services like Google Street View.

The structure of the rest of this chapter is as follows: in Section 2.2, we discuss related work in road and terrain classification. Subsequently, Section 2.3 contains the description of the proposed system for road type classification based on accelerometer and visual features. Next, Section 2.4 details a method for road/terrain classification by unsupervised learning of image features. The process of data collection, ground truth creation, and the evaluation results are presented in Section 2.5. Section 2.6 concludes this chapter.

2.2 Related work

To our knowledge, the use of multimodal data for road/terrain classification has not been investigated by current mobile-sensing solutions since the related works in this area rely either upon accelerometer data or on images. Based on the observation that traversing different terrain types induces different vibration signals, Weiss et al. [17] use an accelerometer mounted on an autonomous vehicle to perform vibration-based road classification. To classify the vibration signals they use a set of distinctive accelerometer features to train a support vector machine (SVM), which was shown to outperform alternative classification methods. Although they correctly classify around 80% of the test samples, the speed of the vehicle is slow and the experiments were performed in a controlled environment. The set of accelerometer features, however, is well-chosen and will (partly) be used in our method. A similar SVM-based approach is presented by Ward and Iagnemma [18], where the algorithm is validated with experimental data collected with a passenger vehicle driving in real-world conditions. The algorithm is shown to classify multiple terrain types with an accuracy of 89%. However, they make use of expensive, specialized sensing equipment to achieve this accuracy, and the classifier was only trained to recognize four very distinctive classes. When the

vibration patterns of the classes would be closer to each other, e.g., when comparing tiles to cobblestones and asphalt to gravel, confusion of classes is expected to be higher, leading to lower accuracy. By using visual features, in addition to the accelerometer data, we are able to tackle this problem.

When vibration data from on-board accelerometer sensors or inertial measurement units is not available, visual terrain classification can be used. In road type classification from visual data, Popescu et al. [19] classify road surfaces based on texture features obtained from statistical properties of medium co-occurrence matrices of road images. Tang and Breckon [20] use a feature set of color, texture, and edge features (some similar to ours) from constrained sub-regions of driver's perspective images to train a neural network classifier of road types. For the color features, they derive histogram distributions and pixel statistics (mean, standard deviation, and entropy) from selected channels of different color space representations of the images. The texture features are based on gray-level co-occurrence matrix statistics and Gabor filters, while the edge features are based on Hough line fitting and contour tracking of the Canny edge output of an image. Khan et al. [21] calculate SURF features, over intersections of a regular grid, from terrain images captured by a mobile robot. The extracted features are used to train a random forest (RF) classifier [22] to discriminate between terrain surfaces.

It is important to note here that unlike the majority of related methods, we make use of smartphone camera images or online images which have been captured in fast motion, and therefore contain artifacts (such as motion blur, illumination changes, overexposed areas, ununiform terrain surfaces etc.) which are challenging for terrain classification. Also, since SVMs have been shown to perform well in several related works, in our approach for multimodal terrain classification, we investigated the performance of both SVM and RF classifiers on our experimental data. A gain of 7% was achieved when using RF instead of SVM for the classification based only on visual features. For the classification of the accelerometer features, the gain was lower, however, still 2%. Therefore, we have adopted the RF classifier for our multimodal approach. In Sections 2.3 and 2.4 we describe in detail our proposed methods, which make use of multimodal and visual data respectively.

2.3 Multimodal terrain classification

The multimodal bike sensing system relies on data from three sensors: an accelerometer, a digital camera, and a GPS, which are commonly found in current smartphones. Each of these sensors independently and concurrently captures surface terrain data. Based on the vibra-

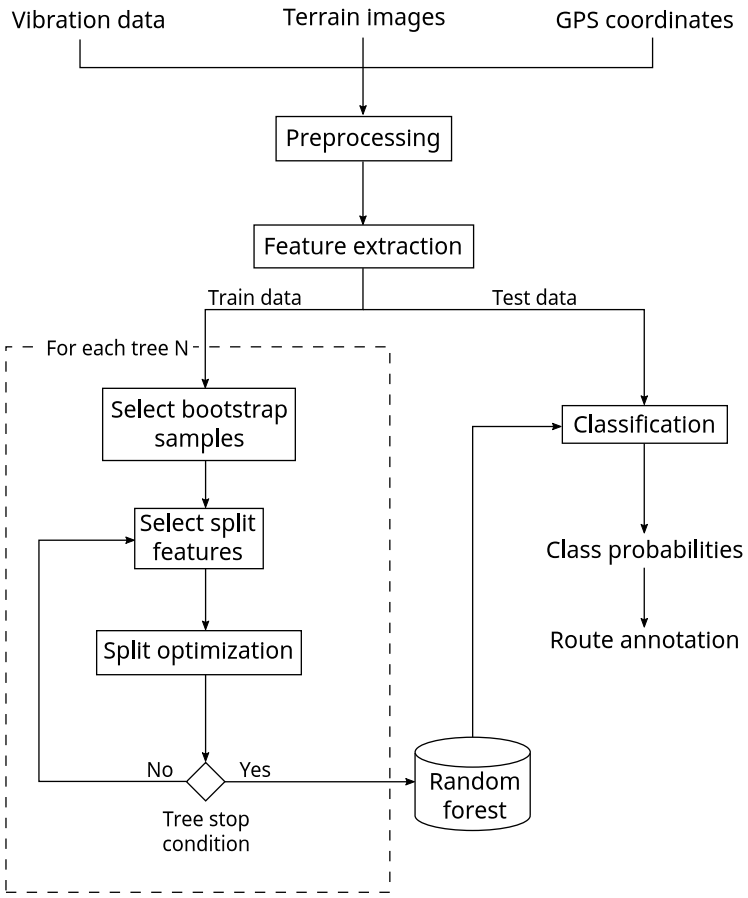


Figure 2.2: Multimodal RF-based classification system.

tion and image data, the proposed terrain classification system tries to predict the type of terrain the vehicle is currently traversing. We differentiate between six terrain classes: asphalt, cobblestones, tiles, gravel, grass, and mud. A general scheme of the classification system is shown in Figure 2.2. First, the raw sensor data is preprocessed. The windowing groups the vibration data into overlapping data frag-

ments of five seconds and aligns them onto the corresponding images and GPS coordinates. The images are also split into blocks in order to detect conflicting or confusing zones, as in the work of Popescu et al. [19]. Subsequently, we calculate a set of feature vectors as described in Section 2.3.1. Next, the feature vectors from the training data are used to train a random forest classifier. Finally, the trained RF classifier is evaluated on features extracted from the test data. Based on the predicted class probabilities and the corresponding GPS data, terrain annotation of novel routes can be performed.

2.3.1 Feature extraction

On every five seconds of cycling, we extract from the sensors' data a set of discriminative visual and vibration features which best describe the road/terrain conditions. The selection of these features is based on a study of the state-of-the-art (discussed in Section 2.2), and on our observations of the collected data. When features showed a similar behavior, the feature with lowest computational cost was chosen.

2.3.1.1 Vibration features

The accelerometer of the mobile device(s) detects the vibration along the X, Y and Z-axes. Depending on the position of the device, the recorded GPS coordinates (x, y, z) will vary and will complicate the classification task. In order to overcome the constraint of forcing the user to place the device in a predefined position, the magnitude m of the acceleration is calculated:

$$m = \sqrt{x^2 + y^2 + z^2}. \tag{2.1}$$

Computing (and analyzing) the features on the vibration magnitude m, instead of on the individual accelerometer data along the X, Y and Z-axes, enables our system to assume a random and possibly changing orientation for the mobile device, i.e., it increases the user's freedom in using the system. This is also important considering different inclinations of the terrain configuration. The set of features which were found to best describe the bicycle vibrations are a combination of the ones proposed in Weiss et al. [17] and Reddy et al. [23], and are defined as follows:

- Mean μ of the acceleration magnitude m – for flatter/smoother surfaces (e.g., asphalt), $\mu(m)$ is low.

- Maximum of the acceleration magnitude m – takes large values for terrain types that contain large surface irregularities, e.g., cobblestones, grass, and mud.

- Minimum of the acceleration magnitude m – takes larger values for flat terrains, such as asphalt.

- Standard deviation of the acceleration magnitude m – is higher for coarse terrain types (e.g., gravel) than for smoother ones (such as tiles and asphalt).

- The acceleration magnitude m – is large if the acceleration is constantly high, as is the case for cobblestones.

- Energy of the acceleration magnitude m, i.e., the sum of the squared DFT component magnitudes [24] of m – has larger values for coarse terrain types, such as grass, mud, and gravel.

It is important to note that each of these vibration features is calculated over a sliding overlapping time window of five seconds, in order to align them with the visual features which are discussed hereafter. A similar windowing approach has been demonstrated to be successful in previous work [25].

2.3.1.2 Visual features

Some of the investigated terrain types are hard to recognize using vibration data (see experimental results in Section 2.5). Since these terrains have similar vibration patterns, it is not always possible to distinguish between their feature values. Visual features can help to overcome these problems. The other way around, vibration features can help to cope with (possible) visual ambiguities. The set of visual features that has been found to be most appropriate for the terrain classification task are based on color, texture, edge, and energy measures. Each of the features are designed to discriminate a certain type or types of road surfaces. We use in total eight features, as follows:

- Color: three features quantifying the percentage of blue, green, and low saturated orange/red pixels in the road image. The features, respectively, give high output for cobblestones and asphalt, grass, and dirt roads and gravel.

- Gray: percentage of pixels that satisfy the RGB color equality R $\approx$ G $\approx$ B. This feature has a higher value for asphalt and cobblestones than for unpaved roads.
- Energy: the Fourier transform energy spread of the road image. The energy is large for road surfaces which contain a lot of edges (such as cobblestones).
- Hough: number of distinct edge directions in the Hough transform of the road image. Road surfaces with structured textures (such as tiles) result in a high number of edges.
- EOH: MPEG-7 edge orientation histogram spread of edges [26]. EOH has large values for road surfaces with random edge distribution (such as grass and gravel).
- GLCM: product of gray-level co-occurrence matrix statistics of local binary pattern filtered road image [27, 28]. High feature values for cobblestones and some unpaved road surfaces.

It is worth noting here that the cycling speed would have an influence on the vibration and, to some extent, on the visual sensor data. A system based on participatory data from many users would invariably encounter data from cyclists moving extremely fast, or slow, however, by utilizing the GPS coordinates of a route such extremes can be easily filtered from the set of data.

After the extraction of the visual and vibration features from the possibly filtered set of sensory data, the feature vectors are divided into training and testing datasets. The training vectors are used to construct a random forest of binary decision trees (discussed hereafter). The test vectors will be evaluated using the trained RF classifier, in order to retrieve the accuracy of the overall terrain classification system.

2.3.2 Random forest classification

Random forest is an efficient classification technique, and provides high classification accuracies [22]. It has shown to be extremely flexible in the context of computer vision [29], and has been successfully applied in practical computer vision tasks, such as human pose recognition in parts from depth images [30]. RF is an ensemble classification method consisting of a number of individual decision trees (see

Figure 2.3). Each of the decision trees in a RF classifier is fitted on a

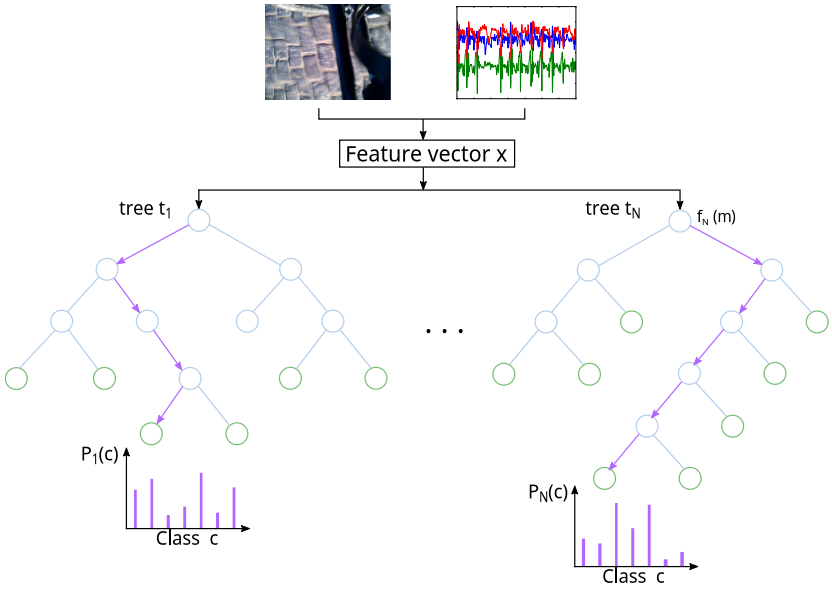


Figure 2.3: Random forest ensemble classification.

bootstrap sample of the training data, i.e., on a part of the training data obtained by sampling with replacement. The leaf nodes of the tree represent a class label, or a value of the target variable. The internal nodes act as split nodes. A random selection of m variables out of all possible M variables is done independently at each split node. A best split function $f(m)$ is then calculated on the selected m variables to separate the samples further down the tree. The computational complexity of training a single binary tree can therefore be given by $O(mn \log n)$, where n is the size of the bootstrap sample. The trees are grown without pruning. Each tree provides a probability distribution over the classes represented in the dataset. The final ensemble probability of the predicted class c, for a feature vector $\mathbf{x}$, can then be calculated as an aggregate of the individual tree predictions in the ensemble:

$$P(c \mid \mathbf{x}) = \sum_{t=1}^{N} P_t(c \mid \mathbf{x}). \tag{2.2}$$

For a more general discussion on random forests, we refer to the work of Breiman [22] and a tutorial of Shotton et al. [31].

2.4 Image-based terrain classification

In Section 2.3, we presented a method for multimodal terrain classification of road types based on features extracted from vibration and visual data. The data used for this method are obtained from a bike-sensing setup (Figure 2.1), for which a common smartphone can be used. However, to capture the terrain of the traversed route with the smartphone's camera, the smartphone has to be positioned in the direction of the road (e.g., mounted on the handlebar of the bicycle). This is somewhat restrictive for the user of the system. To alleviate this problem, available road images from online geographic services can be used (see Section 2.5). In this section, we describe an algorithm for content-based road type classification from images. The proposed algorithm learns road image features from unlabeled samples of different paved and unpaved road images obtained from online services such as Google Street View. In Section 2.5, we will also compare the proposed method with the classification method based only on visual features described in Section 2.3.

2.4.1 Unsupervised learning of road image features

Similar to other feature learning methods, such as the one of Lee et al. [32], our proposed approach learns features from unlabeled images. In particular, we implement a single-layer processing pipeline as the one described by Coates et al. [33]. The processing pipeline consists of two stages: unsupervised feature learning, and feature extraction and classification.

2.4.1.1 Unsupervised feature learning

In the first stage, random patches of size $r \times r$ pixels are extracted from the unlabeled road images, where r is the receptive field size (see Figure 2.4). Each of the extracted patches is reshaped as a vector of pixel values in $\mathbb{R}^m$, $m = r^2 \cdot c$, where c is the number of image channels. Normally, the input images are represented in three-channel RGB color space. However, due to the characteristics of the employed feature learning algorithm, and based on our empirical observations, we introduce a conversion of the input images from RGB to CIELAB [34] color space assuming neutral day illuminant (D65). The transform to a perceptually more uniform color space, such as

CIELAB, enables more accurate distance calculations in algorithms for learning feature mappings from color images. In this way, we construct a dataset $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of randomly sampled patches. Each of the vectors $\mathbf{x}_j \in \mathbb{R}^m$, $j = 1, \ldots, n$ is locally normalized to zero mean and unit variance. Also, the entire dataset of random patches X is whitened [35]. The preprocessed dataset is then used for unsupervised learning of road image features.

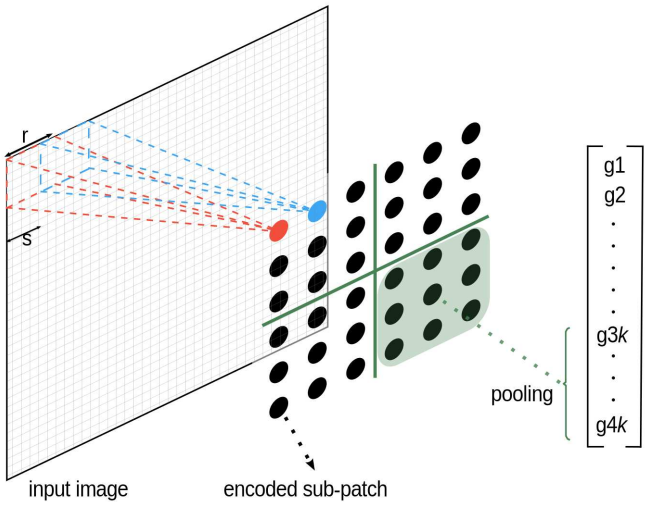


Figure 2.4: Illustration of feature extraction from an input image. First patches of size $r \times r$ are sampled from the image. Each of the patches are sampled s pixels apart. Then, the reshaped and preprocessed vector representing each patch is mapped to a new k dimensional vector (depicted as filled circles) by using the learned dictionary. Finally, the encoded vectors are pooled over a two-dimensional grid and concatenated to form the final feature vector of the input image.

Experimental results [33] have demonstrated that an over-complete dictionary for feature mapping can be learned effectively with fast unsupervised learning algorithms such as k-means learning. Here, we implement a modified version of an efficient stochastic gradient descent k-means algorithm proposed by Sculley [36]. K-means can be viewed as a greedy algorithm for partitioning n examples into k clusters, so that it iteratively minimizes the objective function:

$$\sum_{i=1}^{k} \sum_{j=1}^{n} dist(\mathbf{c}_i, \mathbf{x}_j), \tag{2.3}$$

where *dist* denotes the chosen distance measure between a cluster centroid $\mathbf{c}_i$, and a sample $\mathbf{x}_j$. The algorithm is sensitive to the initial values of the centroids, therefore, we use an initialization procedure developed by Arthur and Vassilvitskii [37] to improve the produced results. We will use our modified stochastic gradient descent implementation of k-means again in Chapter 3 for learning a dictionary of centroids or bases, and we refer the reader to Algorithm 1 of that chapter for additional details. Here, in order to make the algorithm more adaptable for parallelization on distributed memory systems, the gradient update is performed with a larger step. That is, instead of performing the gradient update step on each of the random samples in a batch, we calculate an update step once for each of the unique centroids to which the samples in the batch are closest to.

As mentioned before, the output of this unsupervised learning algorithm is a dictionary C of centroids or bases $\mathbf{c}_i$, $i = 1, \ldots, k$ learned from an unlabeled training set. The learned bases are then used to map novel input samples to features in the feature extraction and classification step.

2.4.1.2 Feature extraction and classification

Mapping of input samples is done by using an encoding transform. We employ one of the sparse nonlinear encodings given by Coates et al. [33, 38], which performs a soft assignment for each feature $i = 1, \ldots, k$ of the feature vector $g(x)$:

$$g_i(x) = \max(0, \text{mean}(z) - z_i), \tag{2.4}$$

where $z_i = \|\mathbf{x} - \mathbf{c}_i\|_2$. The function in Equation 2.4 produces non-zero values only for the features i where the distance of $\mathbf{x}$ to $\mathbf{c}_i$ is below the average of the distances of $\mathbf{x}$ to $\mathbf{c}$, $\forall \mathbf{c} \in \text{C}$.

The learned feature mapping function $g : \mathbb{R}^m \rightarrow \mathbb{R}^k$ allows for feature extraction from a single $r \times r$ patch. To extract features from a road surface image, we apply the feature extraction over the entire input image. The sampling of the input is convolutional (as shown in Figure 2.4), but it can also be performed with a step-size s between two consecutive patches.

Each of the extracted patches is represented by a vector in $\mathbb{R}^k$ after encoding. Grid regions in the $\mathbb{R}^k$ feature space are averaged to reduce the dimensionality of the feature representation of the input

image, and to improve the robustness of the averaged feature vector to small spatial changes in the image. The averaged, or pooled, vectors are then concatenated into the final feature vector.

For each of the labeled images in a training set, we apply the previously described feature extraction process. The resultant feature vectors and training labels are then used for classification. Because of the large amount of features obtained through unsupervised feature learning, we can make use of a linear classification algorithm. A linear l2 SVM [39] compared favorably to other classification methods. Hence, we trained a linear l2 SVM for classification using cross-validation to determine the regularization parameter of the linear model.

2.5 Datasets and experimental results

In this section, we show experimental results for the proposed terrain classification methods, and describe the datasets used in our experiments.

2.5.1 Multimodal terrain dataset

We have performed several bike tours, and annotated the data to be able to evaluate the proposed multimodal terrain classification system. The data collection was performed using standard 26" and 29" mountain bikes. Multiple cycles with varying terrain conditions (in type and frequency) were performed in several rural and (sub)urban regions all over Belgium. An exemplary run, in which all six terrain types occurred, is shown in Figure 2.5. In order to have varying weather conditions, the cycle runs were spread over the year, both in winter and summer on sunny and rainy days. Furthermore, tyre pressure and tyre types were changed in-between several runs in order to cope with the tyre-vibration dependency. To collect the vibration, visual and GPS data we used a Sony Ericsson Xperia mini Android smartphone and a Garmin Edge 800 bike GPS. On the smartphone, we ran a custom-built accelerometer data logger and the time lapse Android app, which takes a picture every five seconds. The bike GPS collected all geographical data and bike statistics. Based on the timestamps, which are stored for each sensor reading, the sensor data is aligned. As can be seen in the cycle run in Figure 2.5, it is not always easy to distinguish between some of the off-road types. Sometimes,

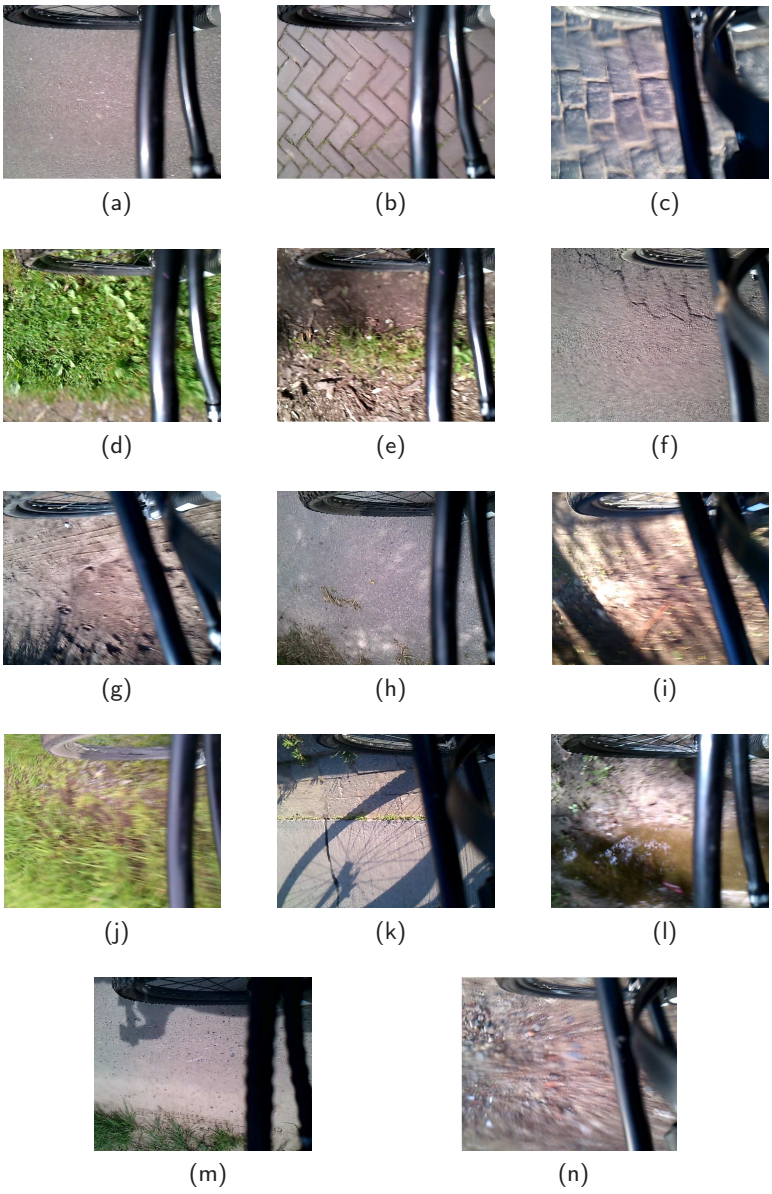


Figure 2.5: Terrains along an exemplary bicycle route of 5.79 km (a)–(n). Terrain classes: asphalt (a), (f), (h), (k), (m); tiles (b); cobblestones (c); grass (d), (j); mud (e), (g), (i), (l); gravel (n).

the terrain consists of a combination of multiple terrain types, e.g., grass and mud, or gravel and mud. For these cases, the annotation task is more challenging and error prone. A similar kind of ground-truth inaccuracy was also reported in Strazdins et al. [40]. Allowing for multi-label classification can be a viable option in improving the system to easily deal with ground-truth annotation errors. Currently, one can also discard these misclassifications from the evaluation metrics. The annotated dataset consists of 240 terrain samples in total, with approximately an equal number of samples for each class.

2.5.2 Multimodal terrain classification experiments

We evaluated the system for both the six-class terrain classification, and also for binary terrain classification: paved versus unpaved terrains. Both of these classification tasks can be found in related work, and depending on the application in which the classification system is used, one can be more relevant than the other. For example, for autonomous vehicle navigation, it might be sufficient to be able to differentiate between paved and unpaved surfaces, while for annotation of recreational cycling routes, a detailed terrain classification is preferred. Like in the work of Khan et al. [41], the evaluation is performed using ten-fold cross-validation. That is, the data is randomly divided into ten equal-sized parts, and each part is used as the test set with training done on the remaining 90% of the data. The test results are then averaged over the ten cross-validation runs. For each experiment, we report the results achieved after a grid-search of the optimal number of trees in the RF classifier, and the split variable ratio.

We evaluated each of the modalities individually, i.e., the classification was performed by using the vibration features and the visual features separately. Afterwards, we combined both modalities for the multimodal classification. For this classification, the strategy that gave the best results was winner-take-all, where the modality with higher class probability determines the final class label. Similar to Ravi et al. [24], we also performed leave-one-out feature evaluation, in order to find out which features among the selected ones are less important than the others. For this, we ran the classification algorithm with one feature variable removed at a time. The acceleration magnitude feature, and the Fourier transform energy spread visual feature turned out to be the least significant. Leaving them out, how-

ever, leads to a significant change of 2-3% in accuracy, i.e., a trade-off between accuracy and computational complexity.

For the six-class accelerometer classification, an accuracy of 71% was achieved by the proposed method. This is the overall accuracy, i.e, the percentage of correctly classified samples out of all samples in the test set. For the binary (paved versus unpaved) classification

Table 2.1: Confusion matrices for visual features (a) and accelerometer features (b).

(a)

	A	C	T	G	M	Gr
A	37	0	0	1	0	0
C	2	38	0	0	0	0
T	1	1	38	0	1	0
G	0	0	0	32	4	0
M	0	0	2	6	34	0
Gr	0	1	0	1	1	40

(b)

	A	C	T	G	M	Gr
A	39	0	0	1	0	2
C	0	35	0	8	6	0
T	0	0	39	6	2	6
G	1	1	1	13	5	4
M	0	4	0	5	21	6
Gr	0	0	0	7	6	22

task, an accuracy of 87% was obtained. For the classification based on visual features only, accuracies of 90% and 96% were achieved for the six-class and the binary terrain classifications, respectively. For the classification based on visual features, most classification errors occur between the grass and mud classes (see Table 2.1), while when only vibration features are used, there are also some misclassifications of the cobblestones and tiles samples to grass or mud classes. When using both the vibration and visual features, the accuracy for the six-class terrain classification is almost 92%, while that of the binary terrain classification is 97%. The gain of the multimodal classification is not that big (less than 2%) when compared to using the visual features alone. However, since the positioning of the smartphone camera is optimal in our dataset, the accuracy of the visual analysis may not always be as high.

2.5.3 Online road image dataset

In order to test the possibility of using available online images for road or terrain classification, we have built a dataset of small road surface images from the paved and unpaved terrain classes (see Figure 2.7). The dataset is constructed using the geographical information from trajectories traversed by recreational cyclists in combination with the

Google Street View web service.

In order to sample different road surfaces, we extract geo coordinates (latitude, and longitude) from points along a GPS trajectory. Duplicate trajectory points are removed and are not considered for further processing. To prevent redundant samples of road images in the final dataset, the trajectory points are filtered so that each point is at least 50 meters apart from the previous point. The distance between trajectory points, given their respective latitudes φ and longitudes λ, is calculated using the Haversine equation for the shortest distance d between two points over the Earth's surface:

$$\begin{aligned} a &= \sin\left(\Delta\varphi/2\right)^2 + \cos\left(\varphi_1\right)\cos\left(\varphi_2\right)\sin\left(\Delta\lambda/2\right)^2 \\ d &= 2R\arcsin\left(\sqrt{a}\right), \end{aligned} \tag{2.5}$$

where R denotes the radius of the Earth. Once we obtain the filtered

(a) (b)

Figure 2.6: Google Street View images from the paved (a) and unpaved (b) road classes.

subset of geo coordinates from a given trajectory, we use the Google Street View API[3] to query images from the selected locations.

One issue of the proposed approach for road image querying is how to obtain a good view of the road surface. The Google Street View web service allows for optional parameters in the image query, such as pitch, which specifies the angle of the camera (up or down) relative to the Street View vehicle. A pitch of -90 degrees gives a

[3] `https://developers.google.com/maps/documentation/streetview/`

Figure 2.7: Samples of road surface images from the paved (a) and unpaved (b) road classes. Within a class, there are samples with very different color and texture characteristics (compare surfaces from asphalt roads and the red bicycle lanes in (a)). There are also very similar samples between classes (see patch on third row, second column from (a), and patch on second row, first column from (b)).

camera view perpendicular to the road surface. However, with the camera in a straight down position, the quality of the image obtained is limited (Figure 2.8). The reason is due to the way the camera is mounted on the vehicle, i.e., the image from the road perpendicular camera view has to be interpolated from images taken from different angles of the camera relative to the vehicle. We use instead a different approach to obtain images with a clear view of the road, such as the images in Figure 2.6. Keeping the pitch to 0°, we calculate for each position the compass heading θ of the camera with regard to the next position on the trajectory (as shown in Figure 2.9). The heading is calculated from the latitudes φ and longitudes λ of the two coordinate points:

$$\begin{aligned} a &= \sin(\Delta\lambda)\cos(\varphi_2) \\ b &= \cos(\varphi_1)\sin(\varphi_2) - \sin(\varphi_1)\cos(\varphi_2)\cos(\Delta\lambda) \\ \theta &= \arctan\left(\frac{a}{b}\right). \end{aligned} \tag{2.6}$$

The images obtained in this way are suitable for content-based analysis of road surfaces.

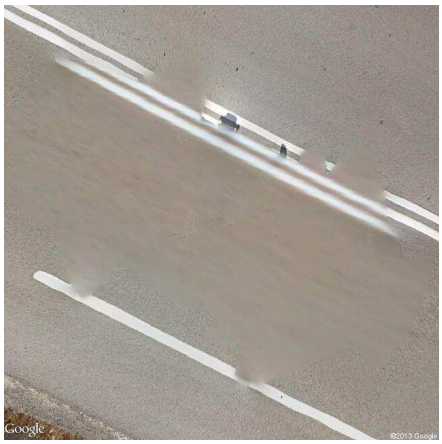


Figure 2.8: Zoomed out Google Street View image perpendicular to the road surface (camera pitch $-90°$). The image contains blurred areas (under the vehicle) where the image content was interpolated.

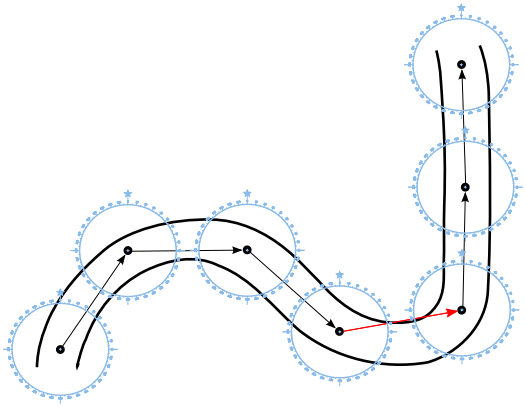


Figure 2.9: Illustration of camera view placement along a trajectory based on compass heading (forward azimuth) calculation between points. The road is in the center of the acquired images. This is not the case (depicted by the red arrow) for only a small number of the acquired Google Street View images, where there is a sharp turn in trajectory direction.

We manually extract 32×32 sub-images from the acquired road images to build our dataset (see Figure 2.10). Because only images from roads traversable by a motor vehicle can be obtained, we create two classes of terrains: paved roads, and unpaved roads. There are in total 20 000 road images in the dataset, where the two classes are proportionally represented by half of the samples. Each of the two classes are comprehensive, i.e. they include samples from different subclasses of road types within the super class. For example, the paved roads class contains sample images from asphalt roads, but also other images of road surfaces with different texture and color, such as cobble stones, tiles, bicycle lanes, pedestrian crossings etc. In the unpaved roads class, there are sample images of different dirt and gravel roads.

Figure 2.10: Extraction of sub-images from the road surface. We extract 32×32 pixel sub-images of different road surfaces to form the road image dataset.

2.5.4 Image-based terrain classification experiments

For our experiments, we used the road image dataset described in the previous section. The dataset was partitioned into a training set of 16 000 images (8 000 images per class), and a test set of 4 000 images (2 000 images for each class). For comparison, we evaluated

both the image-based terrain classification method described in Section 2.4, and the multimodal method from Section 2.3 based only on the visual features. In the latter, we included an additional color feature to quantify the percentage of white pixels in the road images, which improved the discrimination of paved roads containing road signalization.

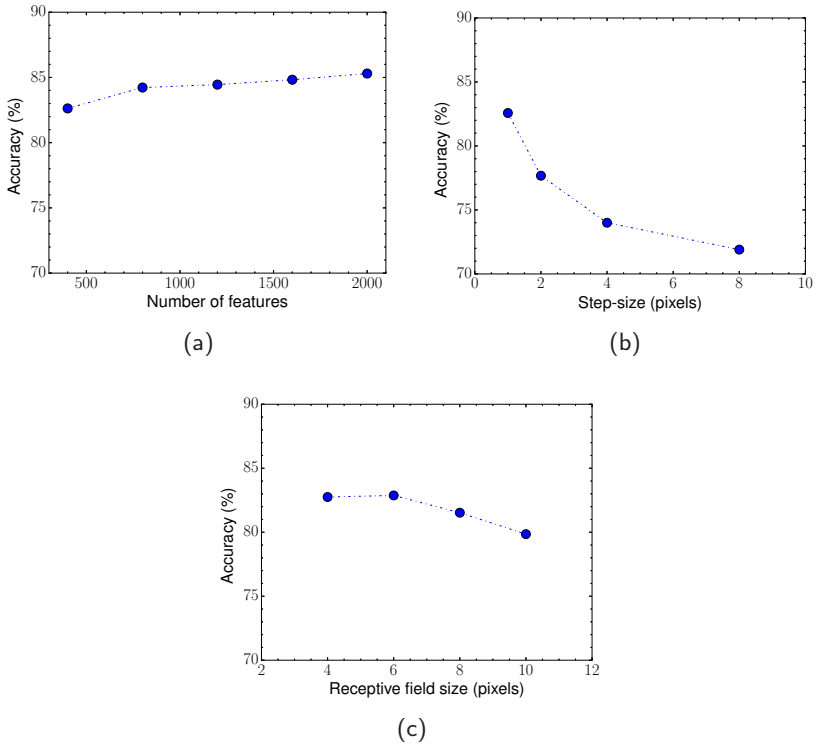


Figure 2.11: Effects of the parameters: number of features (a), step-size (b), and receptive field size (c) on the classification accuracy of the unsupervised road feature learning method.

For each of the compared methods, we used five-fold cross validation to optimize the model parameters. Finally, the learned model was tested on the held out test set. For the unsupervised road image feature learning algorithm, we tested different values for the number of features, the step-size s, and the receptive field r (see Figure 2.11). Because the computational cost prohibits full grid search over all parameters, we varied one parameter while keeping the rest fixed. Af-

terwards, we used the parameter values that achieved the optimal performance for the final test set results (given in Table 2.2).

For the unsupervised feature learning algorithm, when varying the number of features used, better results were obtained when using a higher number of features. As it can be seen in Figure 2.11 (b), convolutional sampling of the input image with a step-size $s = 1$ produced significantly better results than higher step-sizes. For the receptive field parameter, smaller receptive field sizes gave better results. From the experiments, it can be inferred that, except for the step-size parameter, the method is not very sensitive to parameter tuning.

Table 2.2: Test classification accuracy on the online road image dataset.

Method	Test set accuracy
Unsupervised features	85.30%
Engineered features	84.25%

Figure 2.12: Some misclassified samples from the unsupervised feature learning method. Unpaved road samples misclassified as paved road (a). Paved road samples incorrectly assigned to the unpaved road class (b).

2.6 Conclusions and original contributions

In this chapter, we have presented a novel system for multimodal road/terrain classification. The proposed system operates on vibration and visual data to discriminate between different terrain types. Experiments were conducted on classifying between six terrain classes: asphalt, cobblestones, tiles, grass, mud, and gravel, and also on the binary problem of classification between paved and unpaved road surfaces. Experimental results using data obtained from our bike-sensing setup showed that the system yields high classification accuracies, and can be used for automatic route annotation.

We have also presented an image-based terrain classification method, which is based on unsupervised feature learning from road/terrain images available online. For the evaluation of this method, we constructed a challenging road image dataset of 20 000 samples from online images obtained from Google Street View. The experimental results on the comprehensive paved and unpaved road classes from this dataset showed that the proposed method is on par with the multimodal terrain classification system when using only visual features.

Although retraining a domain engineered system, such as our multimodal classifier, can be done just by using a new dataset, the predictive power of the system is coupled with the expert knowledge embedded in its features. Often, however, when training on new data, a revision or modification of the employed features is necessary. On the other hand, the feature learning terrain classification method requires less domain knowledge and more data, and therefore, it is suitable for use in more content-adaptive computer vision systems, such as systems for robot/vehicle navigation.

The work presented in this chapter has lead to the following publications:

- Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, Jürgen Slowack, and Rik Van de Walle. Multi-modal bike sensing for automatic geo-annotation: geo-annotation of road/terrain type by participatory bike-sensing. *10th International Conference on Signal Processing and Multimedia Applications, Proceedings, 2013, pp. 39–49.*

- Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van

Hoecke, and Rik Van de Walle. Image-based road type classification. *Pattern Recognition (ICPR), 22nd International Conference on, Aug 2014, pp. 2359–2364.*

- Steven Verstockt, Viktor Slavkovikj, Pieterjan De Potter, and Rik Van de Walle. Collaborative bike sensing for automatic geographic enrichment. *IEEE SIGNAL PROCESSING MAGAZINE, 2014, vol. 31, no. 5, pp. 101–111.*

Chapter 3

Unsupervised Feature Learning for Hyperspectral Image Classification

In this chapter, we propose an unsupervised feature learning method for classification of hyperspectral images. The proposed method learns a dictionary of sub-feature basis representations from the spectral domain, which allows to effectively exploit the correlated spectral data. The learned dictionary is then used in encoding convolutional samples from the hyperspectral input pixels to an expanded but sparse feature space. Expanded hyperspectral feature representations enable linear separation between object classes present in an image. To evaluate the proposed approach, experiments on several commonly used hyperspectral image datasets are performed. Our experimental results show that the proposed method compares favorably to other pixel-wise classification methods which make use of unsupervised feature extraction approaches. Additionally, even though our approach does not use any prior knowledge, or labeled training data to learn features, it yields either advantageous, or comparable results in terms of classification accuracy to recent semi-supervised methods.

3.1 Introduction

Advances in imaging technology have led to the development of imaging spectroscopy sensors capable of acquiring hyperspectral imagery

with high spatial resolution. Unlike regular three band RGB pictures, hyperspectral images (HSIs) consist of hundreds of spectral bands covering a large interval (e.g., the solar reflective wavelengths 0.4–2.4 μm) of the electromagnetic spectrum. Due to the vast data quantities, and high spectral resolution (at around 10 nm), hyperspectral images offer a large potential for object recognition and classification, thus enabling different geological, agricultural, environmental, and land survey applications. Hyperspectral image classification algorithms, however, are affected by the Hughes phenomenon [5], i.e., the generalization of the learning algorithm is poor since the number of training samples needed to fill the high-dimensional spectral data space is limited.

There are mainly two ways to create labeled training data of hyperspectral images. The first is through field surveys, which provide accurate data, but are expensive and time consuming. The second involves manual labeling of samples through visual recognition. In this method, experts provide the ground truth labels with the aid of digital elevation maps. However, the spatial resolution of the images has to be high to be able to discern the different classes represented in a hyperspectral image. Considering that both ground truth collection approaches rely largely on manual annotation, and require expertise, the availability of labeled training samples remains limited. This renders classification of hyperspectral data ill-posed, which makes full utilization of information present in hyperspectral images challenging.

On the other hand, unlabeled hyperspectral data are plentiful, and can be used in unsupervised learning algorithms for HSI analysis. Therefore, in this chapter we focus on hyperspectral classification methods which are able to effectively exploit the correlated spectral data in an unsupervised or semi-supervised learning manner. In particular, we propose a method for unsupervised learning from subsets of spectral data, which we refer to as spectral sub-feature learning. This is dissimilar to learning dictionaries of spectral pixels, which has been proposed in some supervised hyperspectral classification methods [42–44]. We evaluate our method on pixel-wise HSI classification, and present experimental results on commonly utilized hyperspectral remote-sensing scenes (acquired on multiple sites and by different imaging spectroscopy sensors). When compared to the state-of-the-art unsupervised and semi-supervised techniques, the proposed

method contributes to the improvement of unsupervised HSI classification accuracies, yielding results that are commensurate to the ones achieved by recent semi-supervised methods.

The structure of the remainder of this chapter is as follows: in Section 3.2 we review related work on unsupervised and semi-supervised hyperspectral image classification, and point out differences to our approach. In Section 3.3, we describe in detail our proposed spectral sub-feature learning method. Experimental results performed on five public HSI datasets, which are described in Section 3.4, are presented and discussed in Section 3.5. Finally, Section 3.6 concludes the chapter.

3.2 Related work

A number of approaches in HSI classification focus on feature extraction or feature selection as a way to mitigate the curse of dimensionality inherent in hyperspectral data. The main goal of these dimensionality reduction methods is compression or projection of the data into a lower-dimensional subspace, such that the intrinsic characteristics of the manifold embedded in the high-dimensional hyperspectral data space can be easily discovered. Various unsupervised [45–56], and semi-supervised [57–61] feature extraction methods for the purpose of HSI classification have been proposed in the literature.

3.2.1 Unsupervised methods

Some of the well-known unsupervised feature extraction methods for hyperspectral images are based on principal component analysis (PCA) [45, 46], independent component analysis (ICA) [47], and discrete wavelet transforms (DWT) [48]. PCA-based methods project the hyperspectral data points onto a lower-dimensional orthogonal subspace that best preserves their variance as measured in the high-dimensional hyperspectral data space. On the other hand, ICA-based methods transform the data to independent components by maximizing non-Gaussianity of the components. In contrast to the two previous techniques, wavelet transforms decompose hyperspectral data into high and low frequency features using fixed bases.

Although the former methods provide effective data reduction, the global transformations they produce are often insufficiently flexible

to represent the local information content present in hyperspectral images. For example, PCA is not able to discover non-linear degrees of freedom in the hyperspectral data space, and ICA is less effective when the number of different classes present in an image is large [47].

In recent years, unsupervised learning methods for dimensionality reduction, such as isomap [49], locally linear embedding (LLE) [50], Laplacian eigenmaps (LE) [51], and local tangent space alignment (LTSA) [52] have been proposed in the literature. These methods can estimate the intrinsic geometry of a nonlinear manifold embedded in a high-dimensional input data space by preserving the local geodesic structure of the data points in the dimension reduced space. Such locality-preserving embedding methods exploit a fundamental property of manifolds, namely, that sufficiently small manifold regions are locally linear. This allows any point to be reconstructed as a linear approximation (through a linear combination [50, 51], or local tangent space [52]) of its neighbors. The reconstruction is invariant to neighborhood-preserving transformations and is assumed to be the same in the dimension reduced space. In this way, once the reconstruction weights [50, 51] or local tangent coordinates [52] are calculated in the high-dimensional space, they can be used to calculate the coordinates of data points by minimizing the reconstruction cost in the reduced dimension space. Similar dimensionality reduction methods, such as neighborhood preserving embedding (NPE) [53], locality preserving projection (LPP) [54], and linear local tangent space alignment (LLTSA) [55], which are linear approximations to LLE, LE, and LTSA, respectively, have been used for feature extraction in hyperspectral images [60, 62].

In comparison to other unsupervised feature extraction methods described earlier, our proposed method does not reduce the dimensionality of the HSI data. In contrast, we learn discriminative features by mapping in an expanded but sparse feature space, which allows for linear separability of the classes present in the hyperspectral image.

In cases when class labels are available, better HSI classification results are reported in the literature from the use of semi-supervised methods. Therefore, we also briefly review some of the state-of-the-art semi-supervised learning approaches.

3.2.2 Semi-supervised methods

Compared to unsupervised learning approaches, semi-supervised methods for HSI classification make use of only limited labeled data together with unlabeled data. Due to high costs of creating labeled datasets, as well as efficacy, which can be on-par with that of supervised techniques, semi-supervised learning methods are a viable option for real-world HSI classification applications. Representative semi-supervised methods applied to classification of hyperspectral images include semi-supervised discriminant analysis (SDA) [57]. SDA uses a graph Laplacian-based regularization constraint in linear discriminant analysis (see Chapter 4, Section 4.3 for a description of LDA) to include local manifold information from unlabeled samples and prevent over-fitting when there are insufficient labeled data. Semi-supervised local Fisher discriminant analysis (SELF), proposed by Sugiyama et al. [59], uses a trade-off parameter on the scatter matrices to linearly combine contributions of a supervised method (local Fisher discriminant analysis [63]) and an unsupervised method (PCA). Liao et al. introduced semi-supervised local discriminant analysis (SELD) [60]. In the same manner as SELF, SELD combines supervised LDA with an unsupervised learning method in the category of local linear feature extraction methods (NPE, LPP, or LLTSA). However, unlike SELF, the combination of the contribution of the supervised and unsupervised learning method is nonlinear and nonparametric. Shao and Zhang [61] use the regularized scatter matrices from SELF into an objective function to combine advantages of SELF with an unsupervised dimensionality reduction method: sparsity preserving projections (SPP) [56]. The contribution of each learning method is then controlled by a trade-off parameter.

In the following section, we will first outline the major aspects of our spectral sub-feature learning method, followed by a description of the parts constituting its inner workings.

3.3 Spectral sub-feature learning

Recently, significant research efforts in machine learning have been concentrated toward algorithms for unsupervised learning of features directly from input data for high-level tasks such as classification and recognition. Much progress has been made in different computer

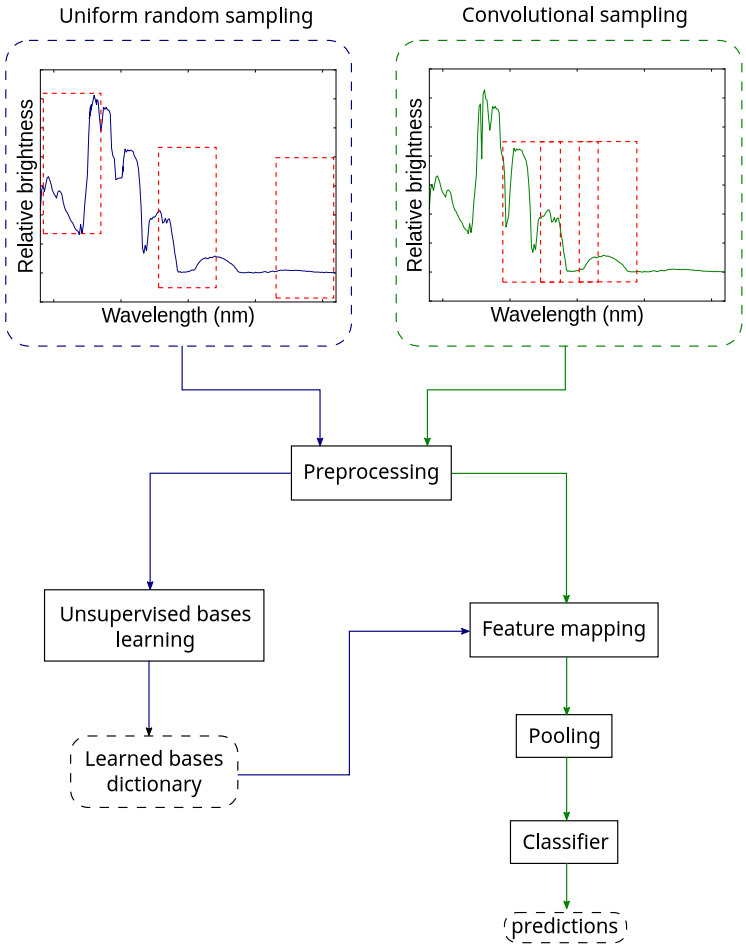


Figure 3.1: Architecture of the proposed spectral sub-feature learning method.

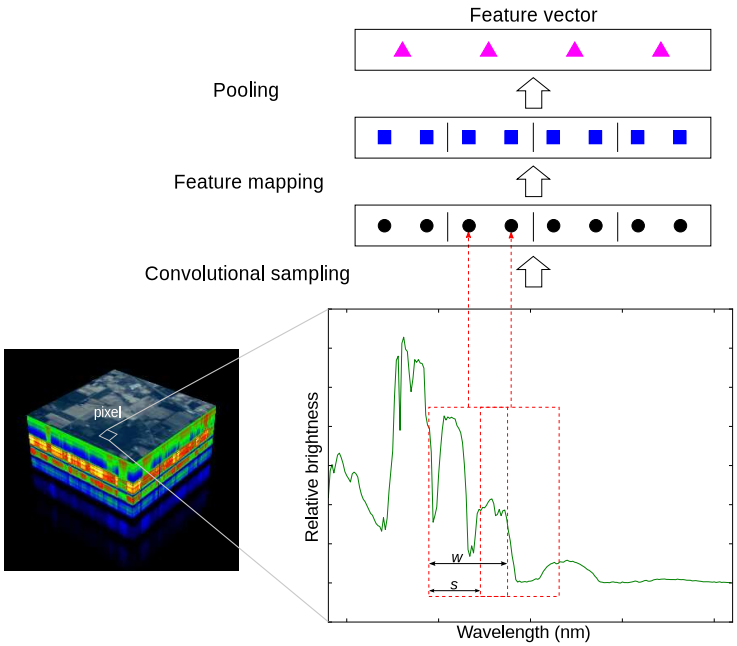


Figure 3.2: Diagram of feature extraction for HSI classification. Convolutional samples are first extracted from an input pixel. The extracted and preprocessed samples (black circles) are then mapped to feature vectors (blue squares) using the learned mapping function $g : \mathbb{R}^w \rightarrow \mathbb{R}^k$, where w denotes the width of the convolutional sampling window, and k is the dimensionality of the mapped feature vectors (blue squares). The mapped feature vectors are then locally pooled in blocks. Pooled feature vectors from each block (magenta triangles) are concatenated to obtain the final feature vector.

vision tasks [32,33,38,64–66] with models trained on datasets consisting of grayscale or color images. Inspired by the good performance of these feature learning systems for generic visual object classification, we propose a method for learning discriminative features from subsets of hyperspectral bands. Unsupervised feature learning methods for visual object classification typically rely on large sets of (single channel or RGB) images for training purposes. However, hyperspectral datasets normally consist of a single acquisition of a scene containing large number of channels, which renders them incompatible with the previously mentioned feature learning approaches. Therefore, the proposed algorithm operates directly in the spectral domain, scaling to a large number of spectral bands. Furthermore, by effectively incorporating the information from all available bands, the problem of selection of optimal bands is eliminated.

The building of our feature learning model can be summarized in two stages. In the first stage, during training, we learn a dictionary of basis vectors using an unsupervised learning algorithm. For comparison purposes, two different algorithms were used for learning the basis vectors—sparse modeling and a fast stochastic gradient descent variant of k-means clustering. In the second stage, the learned basis vectors are applied to map convolutional samples from the spectral pixel space to the feature space with the help of an encoding function (see Figure 3.2). Encoded samples are further pooled, which reduces the dimension of the final feature vector. Lastly, we train a linear classifier using the newly obtained feature vectors in order to be able to predict the class of each hyperspectral pixel. In this section, we will describe our proposed approach in more detail.

3.3.1 Sampling and preprocessing

We start by sampling adjacent hyperspectral sub-bands uniformly at random with a window of width w from unlabeled hyperspectral pixels. Each sample represents a vector in $\mathbb{R}^w$. In this way, we construct a dataset $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ used for learning basis vectors. Each of the vectors $\mathbf{x}_j \in \mathrm{X}, j = 1, \ldots, n$ is locally normalized to zero mean and unit variance, and the entire dataset X is whitened [35] to decorrelate the data. The preprocessed dataset X is then used as input to the unsupervised bases learning algorithm.

3.3.2 Unsupervised learning

The goal of the unsupervised learning algorithm is to learn a dictionary of basis vectors, such that a feature mapping function $g : \mathbb{R}^w \rightarrow \mathbb{R}^k$ can be found, which maps an input vector $\mathbf{x}$ to a new feature vector $\mathbf{y} = g(\mathbf{x})$. We compare two different unsupervised learning algorithms for learning an over-complete dictionary: one based on a sparse modeling approach, and the other based on an efficient stochastic gradient descent k-means algorithm.

3.3.2.1 Dictionary learning via sparse modeling

The fundamental idea of sparse modeling is learning a dictionary of basis vectors, so that novel input data can be represented as a sparse linear combination of the dictionary elements. If $\mathbf{D} \in \mathbb{R}^{w \times k}$ is a dictionary with k basis elements as columns, $\mathbf{A} \in \mathbb{R}^{k \times n}$ represents the sparse decomposition coefficients, and $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{w \times n}$ is a training dataset, then sparse modeling can be represented as a joint optimization problem with respect to $\mathbf{D}$ and $\mathbf{A}$:

$$\begin{aligned} \min_{\mathbf{D},\alpha} \quad & \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \left\| \mathbf{x}_i - \mathbf{D}\alpha_i \right\|_2^2 + \lambda \left\| \alpha_i \right\|_1 \right) \\ \text{s.t.} \quad & \forall \mathbf{d}_j \in \mathbf{D},\, j = 1, \ldots, k, \quad \mathbf{d}_j^{\mathrm{T}} \mathbf{d}_j \leq 1. \end{aligned} \tag{3.1}$$

In (3.1), an l_1 penalty is introduced on the decomposition coefficients $\alpha_i \in \mathbf{A}$, $i = 1, \ldots, n$, to yield sparse solutions, where λ is a regularization parameter. The optimization problem is convex when either $\mathbf{D}$ or $\mathbf{A}$ are fixed, thus it can be iteratively solved by alternatingly optimizing with respect to one variable (the bases $\mathbf{D}$ or coefficients $\mathbf{A}$) while keeping the other fixed [67]. Because second-order derivative batch optimization methods can be impractical on large datasets, we use an online dictionary learning algorithm based on stochastic approximations proposed by Mairal et al. [68].

3.3.2.2 Mini-batch stochastic gradient descent *k*-means dictionary learning

For the purpose of learning basis vectors from hyperspectral data, we implemented on the graphics processing unit (GPU) a modified version of an efficient stochastic gradient descent (SGD) k-means variant [36]. Because the convergence of k-means clustering is guaran-

Algorithm 1 K-means algorithm with mini-batch stochastic gradient descent cost minimization.

procedure SGD k-MEANS(k, b, t, X)
    Input: dictionary size k, mini-batch size b,
        iterations t, dataset X
    Return: learned dictionary $\mathbf{D}$
    $\mathbf{D} \leftarrow \varnothing$
    ▷ Pick $\mathbf{x} \in$ X uniformly at random
    $\mathbf{x} \leftarrow$ uniform_rand_gen(X)
    $\mathbf{D} \leftarrow \mathbf{D} \cup \mathbf{x}$
    while $|\mathbf{D}| < k$ **do**
        ▷ Pick $\mathbf{x} \in$ X with a probability proportional to $\text{cost}(\mathbf{x}, \mathbf{D}) = \min_{\mathbf{d} \in \mathbf{D}} \|\mathbf{x} - \mathbf{d}\|_2$
        $\mathbf{x} \leftarrow$ weighted_rand_gen(X, $\mathbf{D}$)
        $\mathbf{D} \leftarrow \mathbf{D} \cup \mathbf{x}$
    end while
    $\mathbf{v} \leftarrow 0$ ▷ Per-basis counts
    for $i \leftarrow 1, t$ **do**
        N $\leftarrow b$ random samples from X
        for all $\mathbf{x} \in$ N **do**
            ▷ Cluster $\mathbf{x}$ to nearest basis and cache label
            $\mathbf{l}[\mathbf{x}] \leftarrow f(\mathbf{D}, \mathbf{x})$
        end for
        for all $\mathbf{x} \in$ N **do**
            $c \leftarrow \mathbf{l}[\mathbf{x}]$ ▷ Get current label
            $\mathbf{d} \leftarrow \mathbf{D}[c]$ ▷ Get corresponding basis
            $\mathbf{v}[c] \leftarrow \mathbf{v}[c] + 1$ ▷ Update counts
            $\eta \leftarrow \frac{1}{\mathbf{v}[c]}$ ▷ Learning rate
            $\mathbf{d} \leftarrow (1 - \eta)\mathbf{d} + \eta\mathbf{x}$ ▷ Gradient step
        end for
    end for
    return D ▷ Return the dictionary
end procedure

teed only for a local optimum of its cost function, the clustering result is sensitive to the manner of initialization. Therefore, we employ an initialization procedure proposed by Arthur and Vassilvitskii [37]. The goal of the initialization algorithm is to select each of the initial basis vectors from a different cluster. In order to do so, the initial basis vectors are chosen one at a time, at random, from the dataset with probability proportional to the minimum distance of the basis vectors which are already chosen. The modified dictionary learning algorithm is given by Algorithm 1. Note that the computational complexity of Algorithm 1 is given by $O(nd)$, where n denotes the number of training samples, and d is the number of basis vectors. Such a computational complexity can be prohibitively costly for a large number of training samples, however, by using stochastic gradient learning with mini-batches limits the number of training samples n, which enables training on large datasets.

3.3.3 Feature mapping, pooling, and classification

Using one of the two previously described unsupervised learning algorithms on an unlabeled training set yields a dictionary of basis vectors that can be used to map novel input samples to feature space. In the case of sparse modeling, the feature space is formed by the obtained sparse decomposition coefficients for the set of input data directly after pooling, which is described below. In the case of the stochastic gradient descent k-means algorithm, we employ a sparse non-linear encoding transform given by Coates et al. [38], as it performs a soft assignment of each of the features m of the feature vector $\mathbf{y} = g(\mathbf{x})$:

$$g_m(\mathbf{x}) = \max(0, \text{mean}(\mathbf{z}) - z_m), \tag{3.2}$$

where $z_m = \|\mathbf{x} - \mathbf{d}_m\|_2$, and $\mathbf{d}_m$ is the m-th basis vector in the learned dictionary $\mathbf{D}$. In both cases, the feature mapping function transforms an input sample $\mathbf{x} \in \mathbb{R}^w$ to a new sample in feature space $\mathbf{y} = g(\mathbf{x}) \in \mathbb{R}^k$.

To obtain the final feature vector for a given hyperspectral input pixel, we first convolutionally sample the pixel's hyperspectral bands with a window of width w at a step-size s (see Figure 3.2), and preprocess the extracted samples by employing the same preprocessing transforms described in Section 3.3.1. Then, the extracted samples are mapped using the learned feature mapping. Finally, the mapped

feature vectors are pooled. We perform pooling by averaging blocks of the adjacent mapped feature vectors and concatenating the result. The pooling step allows for reducing the dimension of the final feature vector, and for increasing the robustness of the representation to noise in the spectral reflectance data.

We apply our feature extraction approach on a subset of labeled pixels from a hyperspectral dataset and use the obtained features to train a classifier. Because an over-complete dictionary of basis vectors can be learned through unsupervised learning, we can make use of a linear classifier on the already expanded feature vector representations. Therefore, in all performed experiments, we trained a linear l_2 SVM using cross-validation to determine the regularization parameter of the linear model.

3.4 Hyperspectral image datasets

For our experiments, we utilize five commonly used HSI datasets acquired with different imaging spectroscopy sensors, and on different sites: Kennedy Space Center [69], Indian Pines [3], Washington DC Mall [3], Okavango Delta, Botswana [69], and University of Pavia (Figure 3.3).

Indian Pines is a dataset of a mixed forest and agricultural site in Northwest Indiana. It was acquired in June 1992 by using NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), mounted on an aircraft and flown at about 20 km altitude. It contains 220 spectral bands in a wavelength range from 0.4 to 2.5 μm, with a spectral resolution of 10 nm, and a geometrical resolution of approximately 20 m per pixel. The whole scene consists of 145 $\times$ 145 pixels, and there are 16 land-cover classes.

Kennedy Space Center (KSC) data (614 $\times$ 512 pixels) was captured over the KSC site, Florida, in March 1996 in the scope of NASA's AVIRIS project. It consists of 224 bands with a spectral resolution of 10 nm in the wavelength range from 0.4 to 2.5 μm. Only 176 bands are used in the analysis after removing bands with a low signal-to-noise ratio and water absorption bands. The data was acquired from an altitude of approximately 20 km, and has a geometrical resolution of 18 m per pixel. There are in total 13 classes of various land-cover types identified for classification purposes.

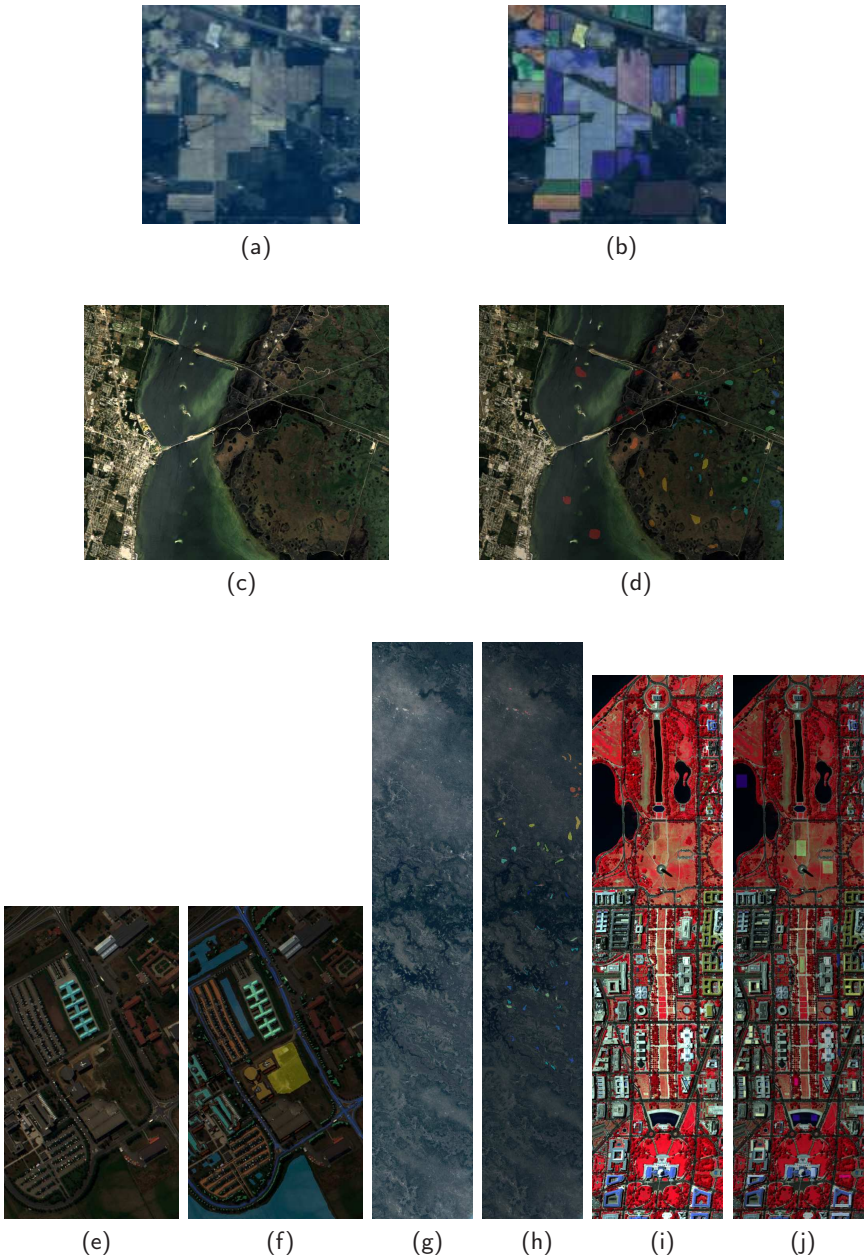


Figure 3.3: HSI datasets. RGB compositions and overlays of the ground truth of the land-cover classes: Indian Pines (a) and (b), KSC (c) and (d), University of Pavia (e) and (f), Botswana (g) and (h), Washington DC Mall (i) and (j).

Table 3.1: HSI dataset classes (Indian Pines and Kennedy Space Center) used in the experiments.

Class #	Indian Pines		KSC	
	Class Name	# Samples	Class Name	# Samples
1	Alfalfa	54	Scrub	761
2	Corn-notil	1434	Willow swamp	243
3	Corn-min	834	Cabbage palm hammock	256
4	Corn	234	Cabbage palm/oak hammock	252
5	Grass-pasture	497	Slash pine	161
6	Grass-trees	747	Oak/broadleaf hammock	229
7	Grass-pasture-mowed	26	Hardwood swamp	105
8	Hay-windrowed	489	Graminoid marsh	431
9	Oats	20	Spartina marsh	520
10	Soybeans-notil	968	Cattail marsh	404
11	Soybeans-min	2468	Salt marsh	419
12	Soybeans-clean	614	Mud flats	503
13	Wheat	212	Water	927
14	Woods	1294		
15	Bldg-grass-trees	380		
16	Stone-steel-towers	95		
Total		10 366		5211

Table 3.2: HSI dataset classes (University of Pavia, Botswana, and Washington DC Mall) used in the experiments.

Class #	Uni. of Pavia		Botswana		DC	
	Class Name	# Samples	Class Name	# Samples	Class Name	# Samples
1	Asphalt	6631	Water	270	Roof	3834
2	Meadows	18 649	Hippo grass	101	Street	416
3	Gravel	2099	Floodplain grasses1	251	Path	175
4	Trees	3064	Floodplain grasses2	215	Grass	1928
5	Metal sheets	1345	Reeds1	269	Trees	405
6	Bare soil	5029	Riparian	269	Water	1224
7	Bitumen	1330	Firescar2	259	Shadow	97
8	Self-blocking-bricks	3682	Island interior	203		
9	Shadow	947	Acacia woodlands	314		
10			Acacia shrublands	248		
11			Acacia grasslands	305		
12			Short mopane	181		
13			Mixed mopane	268		
14			Exposed soils	95		
Total		42 776		3248		8079

University of Pavia dataset was collected with the Reflective Optics System Imaging Spectrometer (ROSIS) hyperspectral sensor of the German national aerospace agency. It is an urban scene (340 × 610 pixels) of the campus of University of Pavia. The original data is composed of 115 spectral bands ranging from 0.43 to 0.86 μm with a spectral resolution of 4 nm. Several bands were discarded due to noise, leaving an image with 103 bands in total. The geometrical resolution is 1.3 m per pixel, and there are nine land-cover classes identified on the university campus.

Washington DC Mall dataset (307 × 1280 pixels) is a HSI dataset captured over an urban site. It consists of 210 spectral bands in the 0.4–2.4 μm range of the electromagnetic spectrum, with a spectral resolution of 10 nm and a geometrical resolution of 2.8 m. After removal of water absorption channels, 191 bands were left from the original data. Seven land-cover classes were identified for the classification task.

Botswana dataset (256 × 1476 pixels) was acquired over a 7.7 km strip of the Okavango Delta, Botswana, in May 2001 using the hyperspectral sensor of NASA's EO-1 satellite. It consists of 242 bands in the wavelength range from 0.4 to 2.5 μm. The geometrical resolution is 30 m per pixel, and the spectral resolution is 10 nm. Only 145 bands of the original hyperspectral data were retained after removing uncalibrated and noisy bands. The data consists of observations of 14 identified classes representing different swamp and drier woodlands areas in the delta.

3.5 Experimental results

We used the five HSI scenes described in Section 3.4 for our experiments. To easily compare the effectiveness of the proposed method, we adopted the experimental setup of Liao et al. [60]. Specifically, each dataset was partitioned, by selecting samples uniformly at random, such that 70% of the samples were used for training, and the rest comprised a test set. Only in the case of the University of Pavia dataset, due to the large number of samples compared to the other datasets, we selected 10 000 samples uniformly at random, and partitioned the sub-sampled dataset to 70% training set and 30% testing set. In order to evaluate the method under low number of training samples, we repeated the experiments for each dataset, using 10% of

the samples (per class) for training, and the rest 90% of the samples as a test set.

In both cases, we used the training set data without the labels to train one of the spectral sub-feature learning algorithms. Then, after applying the learned feature mapping, a linear SVM classifier was trained using the transformed training set together with the corresponding labels. Four-fold cross-validation was used to optimize model parameters. Finally, the model with the optimal parameters was evaluated on the test set.

For the two feature learning algorithms, we evaluate the effect of different parameter values on the classification accuracy. Namely, we vary the size of the learned dictionary $\mathbf{D}$, the width w of the convolutional sampling window, the step-size s between consecutive windows, and the number of pooling blocks (Figures 3.4–3.8).

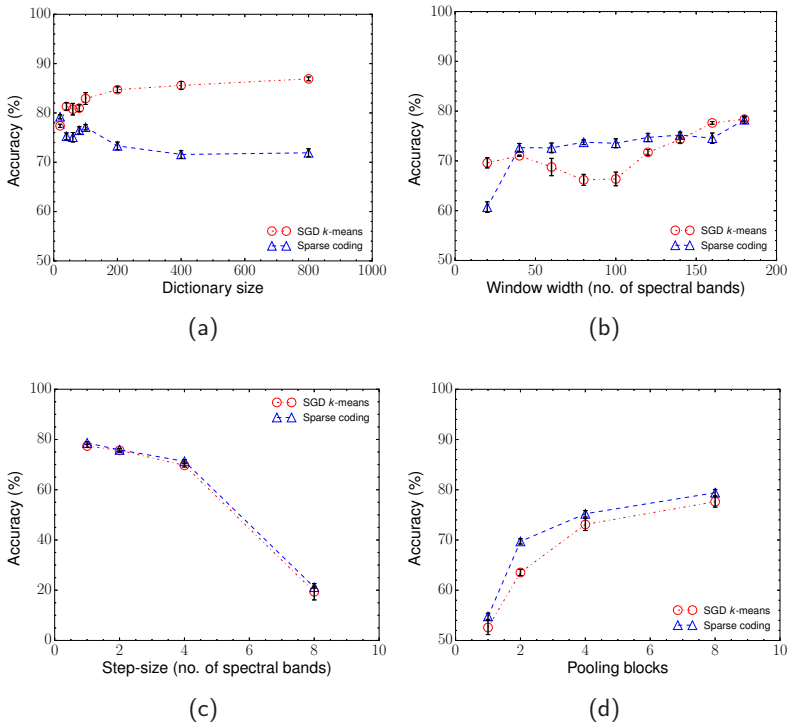


Figure 3.4: Effects of parameters dictionary size (a), window width (b), step-size (c), and number of pooling blocks (d) for the Indian Pines dataset.

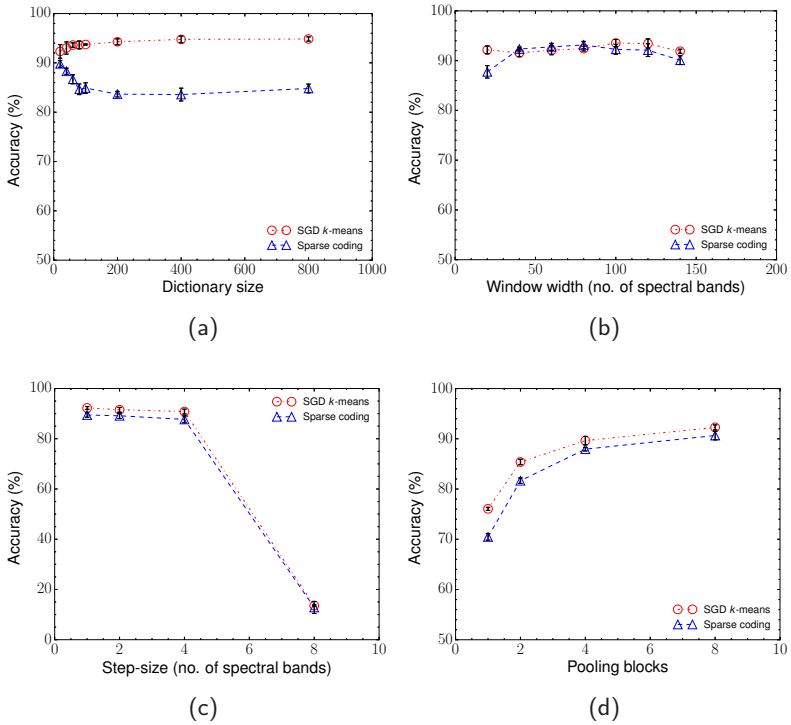


Figure 3.5: Effects of parameters dictionary size (a), window width (b), step-size (c), and number of pooling blocks (d) for the KSC dataset.

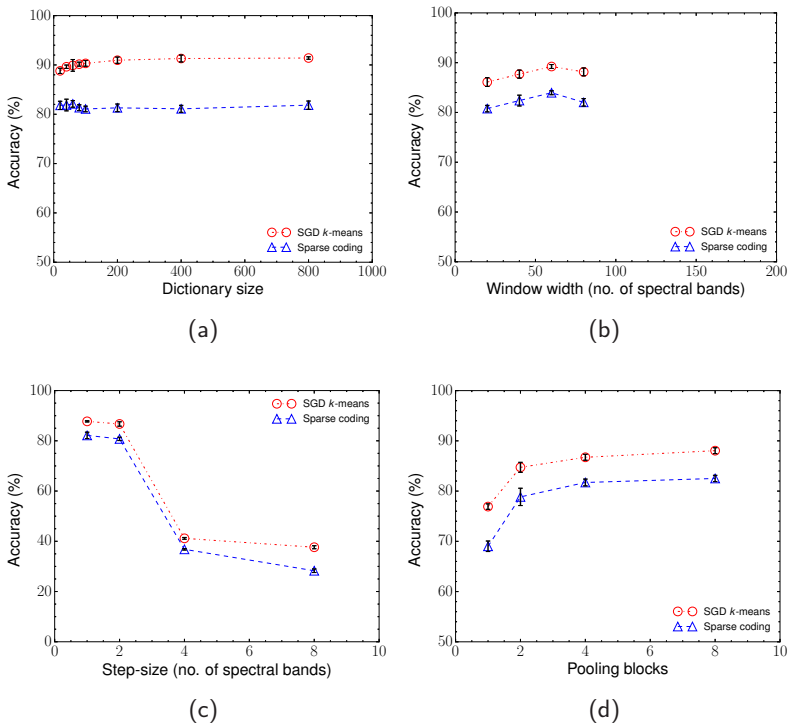


Figure 3.6: Effects of parameters dictionary size (a), window width (b), step-size (c), and number of pooling blocks (d) for the University of Pavia dataset.

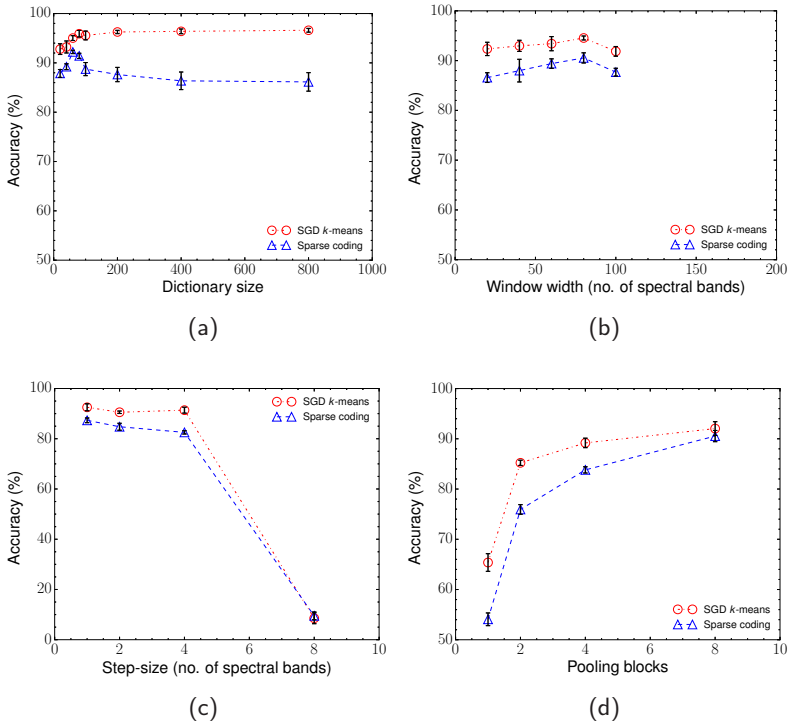


Figure 3.7: Effects of parameters dictionary size (a), window width (b), step-size (c), and number of pooling blocks (d) for the Botswana dataset.

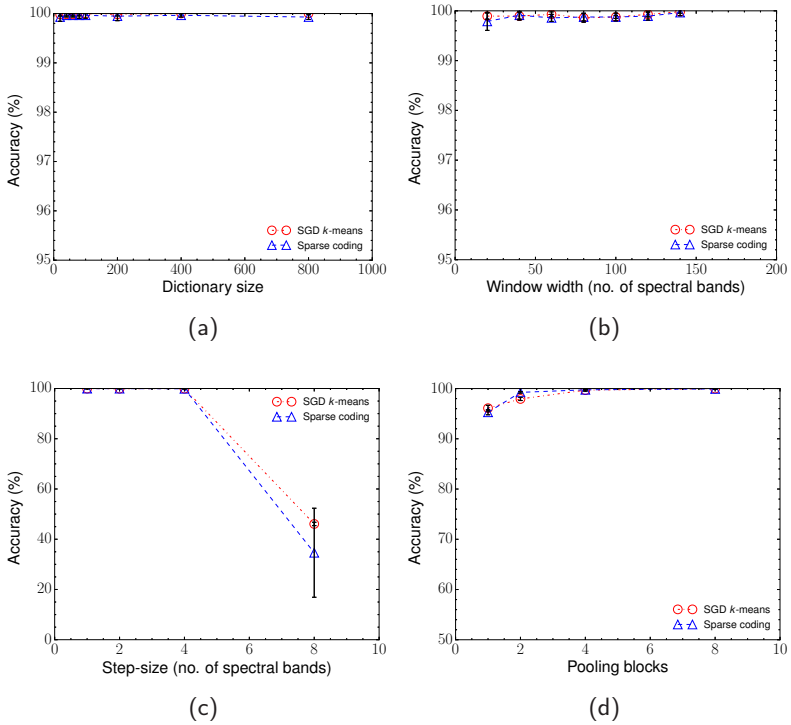


Figure 3.8: Effects of parameters dictionary size (a), window width (b), step-size (c), and number of pooling blocks (d) for the DC Mall dataset.

Table 3.3: Comparison of the overall classification accuracy (%) of different methods for HSI classification. For each dataset, 70% of the samples are used for training, and the remaining 30% comprise the test set. For the proposed method, the mean overall accuracy over five runs is shown together with the standard deviation, as well as the average classification accuracy.

Feature Extraction	Datasets				
	Indian Pines	KSC	Botswana	DC Mall	Uni. of Pavia
Unsupervised Methods					
PCA	73.6	89.6	94	99.7	/
NPE	75.7	91.6	94.5	99.7	/
LPP	75.1	92	93	99.6	/
LLTSA	75.3	90.8	93.5	99.7	/
Our method (sparse modeling) OA	78.72 ± 0.91	93.89 ± 0.63	89.83 ± 4.67	**99.88 ± 0.08**	85.91 ± 1.67
Our method (sparse modeling) AA	74.92 ± 1	89.92 ± 1.11	90.51 ± 4.52	99.23 ± 0.64	82.64 ± 2.83
Our method (SGD k-means) OA	88.49 ± 0.33	**95.26 ± 0.4**	**95.16 ± 1.12**	**99.88 ± 0.03**	90.35 ± 0.17
Our method (SGD k-means) AA	90.1 ± 0.71	92.47 ± 0.61	95.6 ± 0.96	99.62 ± 0.22	86.59 ± 0.21
Semi-supervised methods					
SDA	65.5	89.8	93.9	99.3	/
SELF	73.6	89.6	94	99.7	/
SELD_{NPE}	79.2	93.6	**95.1**	**99.8**	/

Table 3.4: Classification accuracy (%) of the proposed method when using only 10% of the samples in the dataset for training. The mean overall accuracy and average accuracy over five runs are shown together with the standard deviation. The overall and average classification accuracies are also shown when using the raw spectral data without feature extraction.

Feature Extraction	Datasets				
	Indian Pines	KSC	Botswana	DC Mall	Uni. of Pavia
Raw OA	68.89	80.24	81.32	97.4	77.32
Raw AA	55.82	72.01	80.99	87	58.91
Our method (sparse modeling) OA	74.27 ± 2.77	86.91 ± 3.53	77.07 ± 5.36	99.56 ± 0.21	74.6 ± 1.62
Our method (sparse modeling) AA	64.35 ± 6.35	80.38 ± 4.22	77.38 ± 5.62	96.96 ± 1.3	64.29 ± 2.48
Our method (SGD k-means) OA	78.96 ± 0.82	90.61 ± 0.27	85.01 ± 1.29	99.6 ± 0.1	86.92 ± 0.39
Our method (SGD k-means) AA	71.56 ± 0.93	86.57 ± 0.26	85.99 ± 1.15	96.44 ± 0.93	81.63 ± 0.87

Due to computational constraints of a full grid search over all parameters, each parameter was optimized while keeping the rest fixed. The optimal parameter values were then used to train the final model, which was evaluated on the test set. In the case of the algorithm based on sparse modeling, there is an additional parameter λ, which is the sparsity regularization parameter. Since there is no analytical link between λ and the effective sparsity, we tested different numbers of basis vectors using orthogonal matching pursuit (OMP) [70] within the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 40, 60, 80, 100, 200, 400\}$, and used the optimal parameter for each dataset.

From the experimental results shown in Figures 3.4–3.8, it can be seen that over all datasets, and for both dictionary learning algorithms, better results can be achieved when using dense convolutional sampling (lower step-size), and a higher number of pooling blocks. The convolutional sampling with lower step-sizes allows for a more extensive representation of the input. While pooling is necessary to reduce the size of the final feature vectors, the high number of pooling blocks contribute to preserving the discriminative properties of the features. Therefore, we can make an informed decision for these two parameters without defaulting to cross-validation for their estimation. Considering the effects of the dictionary size parameter, when using the SGD k-means dictionary learning algorithm, larger over-complete dictionaries give better results. On the other hand, when using sparse modeling to learn the dictionary, smaller size dictionaries can perform better than over-complete dictionaries. The window width parameter also appears to require cross-validation. Considering the effects of this parameter, however, we can see that improvements in the results can be achieved when using large convolutional sampling windows relative to the total number of spectral bands available. However, we would like to note that convolutional sampling plays an important role in the accuracy of both dictionary learning algorithms. Namely, when setting the window width to the total number of spectral bands (effectively disabling convolutional sampling), the classification accuracy drops significantly.

Recently, deep learning models [71,72] have been successfully used to learn a hierarchical representation of features, where each additional layer of the model represents a higher level representation of the data. Therefore, we also investigated extensions of our model architecture with several layers, by applying the unsupervised learning

algorithms on the transformed feature representations produced by a previous layer. However, the increase of the classification accuracy that we observed was lower than the standard error of the achieved classification scores, which did not justify the additional computational cost incurred by the deep version of our model.

3.6 Conclusions and original contributions

In this chapter, we have proposed an unsupervised spectral sub-feature learning method for classification of hyperspectral images. Experimental results performed on different HSI datasets have shown that the proposed method compares favorably with other unsupervised approaches in HSI classification. Compared to the best results achieved by unsupervised methods in the experiments, our method yields a maximal improvement of overall classification accuracy of up to 12.79%, and an average improvement over all test datasets of 4.22% (cf. Table 3.3). Additionally, even though our algorithm uses no labeled data or prior knowledge to learn features, it can achieve similar or better performance than state-of-the-art semi-supervised methods on the same task. Even though recent supervised methods for HSI classification generally outperform unsupervised and semi-supervised approaches, our method has demonstrated desirable properties on certain datasets, due to the novel use of spectral information in hyperspectral images. As a result, the combination of supervised methods with our approach would be an interesting venue for future work. Furthermore, the ability of the proposed method to fully exploit the information present in the spectral domain is of importance in hyperspectral data analysis. Namely, for the classification of materials in spectral libraries, where methods that make use of spatial or geometrical context information cannot be applied. Finally, spectral sub-feature learning can be useful in frameworks, such as the one proposed by Li et al. [73], which integrate multiple types of features for classification of hyperspectral images.

The work presented in this chapter is published in *International Journal of Remote Sensing*:

- Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Unsupervised spectral subfeature learning for hyperspectral image classification. *Interna-*

tional Journal of Remote Sensing vol. 37, no. 2, pp. 309–326, 2016.

Chapter 4

Supervised Hyperspectral Image Classification

In this chapter, we propose a novel supervised hyperspectral image classification method. The proposed method is based on a convolutional neural network model, and provides an end-to-end learning approach for classification of hyperspectral images. Using only the hyperspectral input data, our model is able to learn structured features, roughly resembling different spectral band-pass filters. The conducted experiments show that our method can achieve high classification results under a low number of training samples scenario. Additionaly, the applicability of the proposed method extends beyond classification of hyperspectral images, and we will see, in Chapter 5, how a similar model can be used in the context of fault classification for rotating machinery.

4.1 Introduction

In Chapter 3, we focused on unsupervised methods for hyperspectral image classification. When labeled training samples are available, however, better classification results are reported in the literature from supervised learning approaches. Therefore, in this chapter we continue exploring the problem of classifying hyperspectral images, but from the perspective of supervised learning. Supervised learning methods for HSI classification benefit from including additional information, in terms of class labels, in the process of optimization of their corresponding objective function. Nevertheless, they too suffer from

the problems stemming from the high dimensionality of hyperspectral images. Furthermore, in Chapter 3, we have demonstrated a new approach to effectively exploit spectral information, by learning dictionaries of sub-feature basis representations in the spectral domain. In this chapter, we will build upon the gained insights of the HSI classification problem, and incorporate advantages offered by supervised learning, while using a low number of labeled training data.

In recent years, the use of contextual information in HSI classification of remote-sensing scenes has been widely accepted, as experimental results have shown significant improvement in classification accuracy for various classification methods when jointly using spatial and spectral information. There are, in general, three different ways of incorporating spatial information in the HSI classification process. Fixed neighborhood methods use information from a predefined neighborhood set (such as the four, or eight-connected neighbors) of each pixel in the scene. By using morphological filters [74], on the other hand, an adaptive neighborhood approach can be formed by defining the morphological spatial structure to which a pixel belongs. A third approach of including spatial information in the classification process is by relying on information from the corresponding image segmentation of a scene. In this way, the neighborhood of each pixel is defined by the non-overlapping homogeneous region to which it belongs in the segmentation map. This spatial context approach allows for higher level semantic information to be included in the classification process, such as the shape and size of the object to which a certain pixel belongs. However, it is also dependent on the quality of the segmentation maps produced by the image segmentation algorithm, which is important considering that automatic image segmentation remains one of the open problems in computer vision. In the HSI classification method proposed in this chapter, we also address joint spectral-spatial classification of hyperspectral imagery.

The remainder of this chapter is organized as follows. In Section 4.2, we provide some background information on neural network models, which are foundational of our proposed method. Section 4.3 gives a review of related methods for HSI classification. In Section 4.4, we describe in detail the proposed convolutional neural network (CNN) model for HSI classification. For the case of model training with a low number of training samples, we propose an augmentation technique for HSI data, which is detailed in Section 4.5.

Experimental results obtained with the proposed approach are given in Section 4.6. Finally, we draw conclusions in Section 4.7.

4.2 Background

Artificial neural networks (ANNs)[1] have been extensively studied in the literature [75–77]. However, since the HSI classification method proposed in this chapter is based on an ANN model, in this section, we provide a short summary of the existing literature describing the most important aspects of the type of neural networks used in our approach.

4.2.1 Feed-forward networks

The feed-forward neural network is one of the most successfully used machine learning models, drawing its inspiration from the information processing organization of biological nervous systems [76]. A feed-forward neural network consists of layers of nodes or units, comprising an input and an output layer, and typically one or more intermediate layers of nodes denoted as hidden unit layers. The layers are connected by a set of edges (each representing an adaptive weight parameter of the network), such that the joint set of nodes and edges forms an acyclic graph (see Figure 4.1). The output of each hidden unit of the network, known as an activation, is a linear combination of the inputs to the unit. The learnable weight parameters serve as the linear combination coefficients, controling the importance of the inputs. The functional transformations of the basic feed-forward network from Fig. 4.1 can thus be represented in the form

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{4.1}$$

where the input variables $x_1, \ldots, x_D$ are combined into M separate linear combinations by the network's hidden units. The parameters $w_{ji}^{(1)}$ denote the weights assigned to the connections from unit i to unit j, where $i = 1, \ldots, D$, and $j = 1, \ldots, M$, and $w_{j0}^{(1)}$ represent the weights of the bias connections, whose input is fixed constant at

[1] We will also use the term neural network (NN) throughout this text to refer to an ANN.

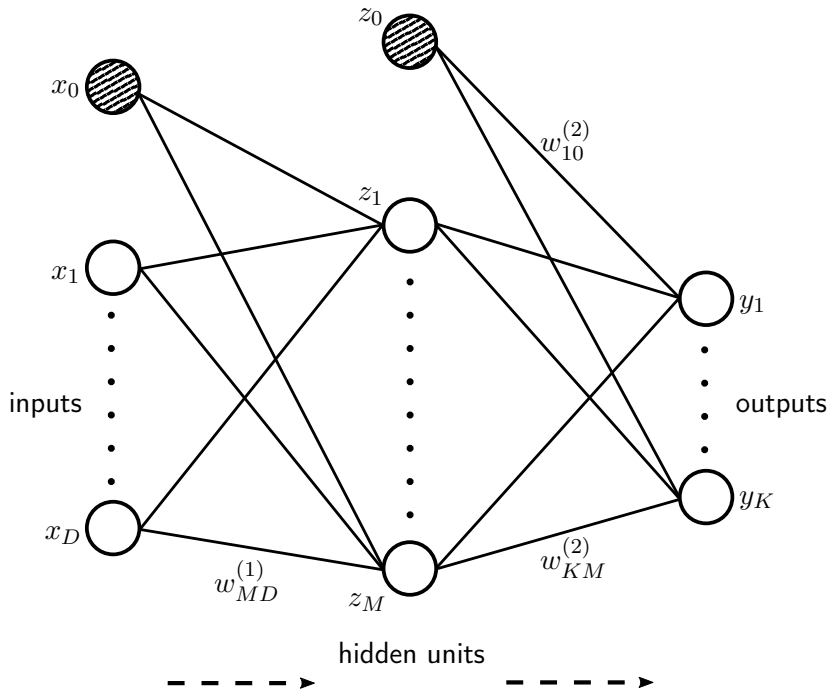


Figure 4.1: A diagram of a two-layer feed-forward neural network. Nodes represent the inputs, outputs and hidden units. The connections between nodes correspond to the learnable weight parameters of the network. The arrows show the flow of information from inputs to outputs during a forward pass of the network. Units x_0 and z_0, which are fixed at a constant output, are connected by the bias weights.

1. The superscript (1) indicates the layer of the network. A feed-forward network, as the one depicted in Figure 4.1, consisting of an input layer, an output layer, and one hidden layer, is considered a two-layer network, corresponding to the total number of layers containing learnable parameters. The activations (4.1) produced by the hidden units are further transformed by a differentiable, nonlinear activation function, to obtain the final hidden units' outputs

$$z_j = h(a_j). \tag{4.2}$$

The activation function $h(\cdot)$ is a nonlinear function, such as the hyperbolic tangent, or the logistic sigmoid, which are differentiable counterparts to the Heaviside function (Figure 4.2). Similarly to (4.1), the

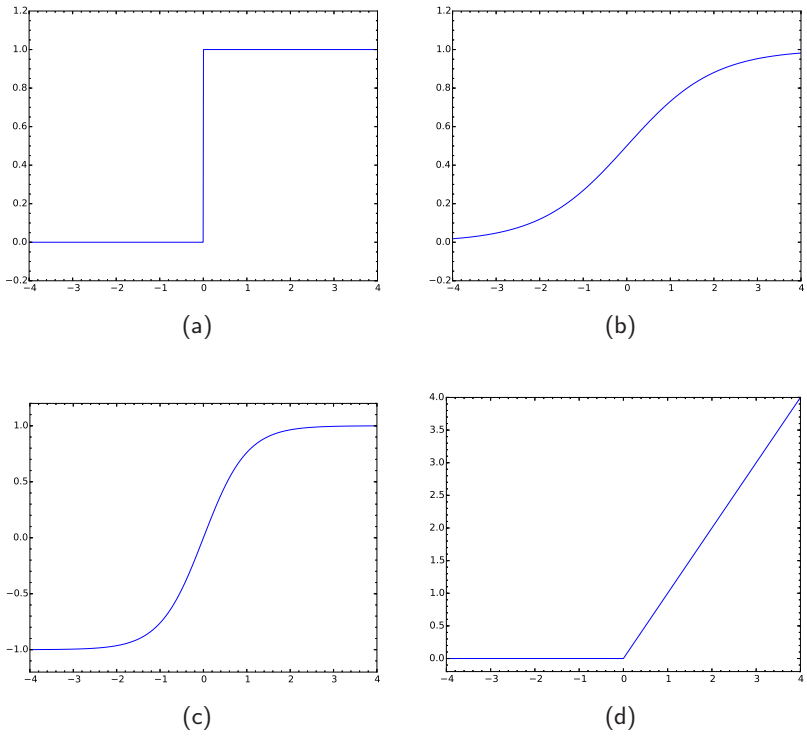


Figure 4.2: Different activation functions. The Heaviside function (a), logistic sigmoid (a), hyperbolic tangent (c), and rectified linear units function (d).

K output unit activations are also obtained as a linear combination of their inputs

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \tag{4.3}$$

where $k = 1, \ldots, K$, and $w_{k0}^{(2)}$ are the bias weights of the second layer of the network. An appropriate activation function is used to obtain the network outputs

$$y_k = \sigma(a_k). \tag{4.4}$$

The choice of the output units' activation function $\sigma(\cdot)$ depends on the distribution of the target variables, and on the nature of the problem. For regression, the identity function is used, so that $y_k = a_k$. For binary classification problems, the logistic sigmoid function is used to transform the output activations so that

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \tag{4.5}$$

On the other hand, the choice of output activation function for multiclass problems is that of the softmax function

$$\sigma_k(\mathbf{a}) = \frac{\exp(a_k)}{\sum_{i=1}^{K} \exp(a_i)}, \tag{4.6}$$

where $\mathbf{a}$ is the K dimensional vector of the output activations $a_1, \ldots, a_K$.

4.2.2 Network training with error backpropagation

Considering (4.1) to (4.4), the output of the network is a nonlinear function $y(\mathbf{x}, \mathbf{w})$ of the input $\mathbf{x}$, and the parameters $\mathbf{w}$. Given a training set of input examples $\{\mathbf{x}_n\}$, where $n = 1, \ldots, N$, and a corresponding set of target variables $\{\mathbf{t}_n\}$, a good way of determining the network parameters is to iteratively minimize an error function dependent on $\mathbf{w}$. In case of regression, and considering a Gaussian distribution of the targets with a mean dependent on $\mathbf{x}$, the error function obtained under a maximum likelihood framework is that of the sum-of-squares error

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2. \tag{4.7}$$

In the multiclass classification case, the corresponding error function is of the form

$$E(\mathbf{w}) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}), \tag{4.8}$$

where the binary target variables $t_k \in \{0, 1\}$ are encoded in a 1-of-K coding scheme indicating the assignment of the inputs to one of the K mutually exclusive classes. Considering the output unit activation function in case of (4.7) to be unity, and in the case of (4.8) to be that of the softmax function (4.6), for a single example $\mathbf{x}_n$ the derivative of the error with respect to the output activation k in both cases results to

$$\frac{\partial E_n}{\partial a_k} = y_k - t_k. \tag{4.9}$$

The rate of change of the error function with respect to the output activations of the network, however, is only one link to understanding how the rest of the network should respond to changes of the error. That is, we are interested in adjusting the weight parameters $\mathbf{w}$ of the network, which can be done with gradient descent, and the error backpropagation algorithm. Since the error function is a nonlinear function of the weight parameters, the goal is to find a vector $\mathbf{w}$, such that the gradient of the error $\nabla E(\mathbf{w})$ vanishes, and $E(\mathbf{w})$ is smallest. This corresponds to a local minimum in the weight space. Gradient descent optimizes the weight parameters by taking small steps opposite the direction of the error gradient at each iteration step τ, such that

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta \nabla E(\mathbf{w}^{(\tau-1)}), \tag{4.10}$$

where the step size is governed by the learning rate η.

The backpropagation algorithm provides an efficient local message passing scheme to compute the gradients of the error. During a forward pass, a unit in a feed-forward network with a general topology computes a linear combination of its inputs, so that

$$a_j = \sum_i w_{ji} z_i, \tag{4.11}$$

followed by a nonlinear activation function transform $h(\cdot)$ to obtain the activation

$$z_j = h(a_j). \tag{4.12}$$

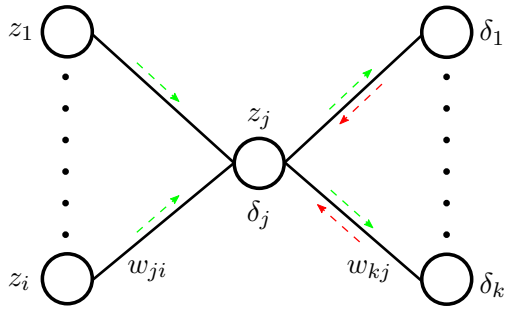


Figure 4.3: A diagram of a hidden unit j from a general feed-forward neural network topology. The hidden unit's δ_j is calculated by backpropagation of the δ's of the units k connected to the outgoing weights of j. The green arrows show the direction of information flow during the forward pass, while the red arrows denote the backpropagation of information.

In (4.11) the biases are incorporated in the sum by including one extra input unit with an activation fixed at 1. The weights w_{ji} connect the input units i with the output unit j. Without loss of generality for batch inputs, the derivative of the single input error E_n with respect to a weight w_{ji} can be represented as

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} \tag{4.13}$$

by applying the chain rule. From (4.11) it follows that

$$\frac{\partial a_j}{\partial w_{ji}} = z_i, \tag{4.14}$$

and if we denote the derivative of the error E_n with respect to the activation a_j as

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j}, \tag{4.15}$$

then (4.13) can be written as

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i. \tag{4.16}$$

From (4.9), we see that we can already calculate the quantity δ_k for an output unit k (Figure 4.3). In the case of hidden units, the δ's can

be calculated as

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j}, \tag{4.17}$$

where k in the sum runs over all units (possibly other hidden or output units) connected to unit j. Considering (4.11) and (4.12) to calculate $\frac{\partial a_k}{\partial a_j}$, and substituting δ_k for $\frac{\partial E_n}{\partial a_k}$ in (4.17) we obtain

$$\delta_j = h'(a_j) \sum_k \delta_k w_{kj}. \tag{4.18}$$

By applying (4.18) recursively in the network, we see that we can calculate the δ for any hidden unit by propagating the δ's of units connected to the output weights of that hidden unit, which in turn enables us to quantify the gradient of the error $\nabla E(\mathbf{w})$.

4.2.3 Convolutional neural networks

The input of a neural network model is usually fixed by the dimensionality of the raw data. Input data such as images, however, exhibit spatial stationarity properties, which result in redundancy of the network input and an increase of the model parameters. Furthermore, since it is not an easy task to develop discriminative hand-crafted features which can succinctly represent the data, we are interested in building models that are robust to small transformations (e.g., rotation, translation, scale) of the inputs. In the case of data represented as images, such transformations pose a significant challenge for computer vision models, even though they are effortlessly handled by the human visual system on daily bases. As such, computational models that incorporate similar invariances in the model structure with respect to their inputs are highly desirable.

Convolutional neural networks have been introduced to take advantage of local pixel correlation in images for the problem of handwritten character recognition [78]. The model draws inspiration from the receptive field cells found in the animal visual cortex [79], which have been found to be sensitive to specific edge-like stimuli. A convolutional layer is organized into planes of units (see Figure 4.4), and the output of these units forms a feature map. Each of the units in a given plane exploits the neighboring pixel correlation by detecting local features from a small patch of the image. By sharing the

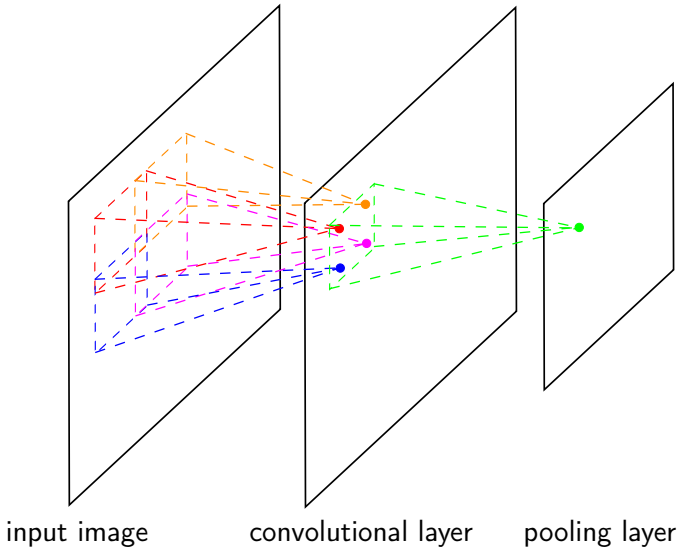


Figure 4.4: A diagram depicting a convolutional layer and a pooling layer of a CNN. The areas marked with dashed lines denote the receptive fields of each layer. The filled circles represent the individual units of the feature maps resulting from the convolution and pooling operations.

weights between units of a plane, all of the units detect the same visual pattern, but at different locations in the image. Also, the shared weights significantly reduce the number of adjustable parameters of the model. Due to the weight sharing, the linear combination of the input pixels with the weights can be implemented as a filtering operation over the input image using the weights as a filter kernel. In practice, multiple weight filter kernels are applied to detect different kinds of elementary patterns, with the end goal of building a hierarchical model which is able to detect higher-order features. In matrix notation, the operation of a convolutional layer can be represented as:

$$\mathbf{X}_m^{(k)} = \sigma\left(\sum_{c=1}^{C} \mathbf{W}_{c,m}^{(k)} * \mathbf{X}_c^{(k-1)} + \mathbf{B}_m^{(k)}\right). \tag{4.19}$$

In Equation (4.19), k denotes the layer of the network, and the $*$ operator is used for the 2D convolution of channel $c = 1, \ldots, C$ of the input $\mathbf{X}^{(k-1)}$ and the filter $\mathbf{W}_{c,m}^{(k)}$, which is responsible for the m-th output feature map $\mathbf{X}_m^{(k)}$, where $m = 1, \ldots, M$. The matrix $\mathbf{B}_m^{(k)}$ contains the bias weights. Finally, a nonlinear activation function σ is applied to the sum of convolutions to obtain the final output.

Since multiple units of a plane in the convolutional layer detect the same pattern in the input image, small translations of the pattern in the image can still be detected by some units, which results in a translation of the output in the corresponding feature map.

The feature maps are then processed by a pooling (or subsampling) layer of the convolutional network. Pooling layers reduce the spatial resolution of feature maps. For example, a 2×2 max pooling layer would produce an output which has twice as few rows and columns as the original feature map, selecting the maximal response of each consecutive four-neighborhood of the feature map. This further reduces the sensitivity of the model to distortions of the input, because each output from the pooling layer corresponds to several receptive field areas (the size of the weight filter kernels) of the input image. Deep CNN models usually consist of several layers of convolutional and pooling layers followed by densely connected layers of hidden units, and corresponding output units.

4.3 Related work

Some supervised feature extraction techniques [80, 81] for hyperspectral images are based on the well-established linear discriminant analysis (LDA) method, which uses labeled samples to find a projection matrix that maximizes the between-class variance to the within-class variance. Another discriminant-based supervised hyperspectral feature extraction method, called nonparametric weighted feature extraction (NWFE) [82], uses an improved discrimination criterion by assigning a higher weight to samples closer to the discrimination boundary region. Kuo et al. [83] proposed a kernel-based hyperspectral feature selection method, which optimizes the linear combination of z-score values of features in the radial basis function (RBF) kernel function.

A sparse modeling dictionary-based approach has been applied in some HSI classification methods [42, 44, 84], where different types of sparsity constraints have been included in the corresponding dictionary modeling cost functions. The general idea of these methods is to learn a dictionary for each class from labeled data [42, 44], or use the labeled data itself to form dictionaries [84], and classify unknown pixels by determining which class specific dictionary best describes the sample in terms of minimum value of the reconstruction error.

ANN based models have also been investigated for the purpose of HSI classification. In the work of Yang et al. [85], an approach inspired by compressive sensing is given, which is based on a single layer feed-forward neural network with sparsity constraints on the input and hidden layer. Deep learning neural network models [86–93], based on CNNs, deep belief networks (DBNs) [64], and autoencoders (AEs) [72], have also been proposed. The latter two types of models learn hierarchical features from hyperspectral input data by greedy layer-wise training of restricted Boltzmann machines (RBMs) [64], or layers of hidden units, followed by supervised training of the whole stacked model for the classification task. The models proposed in the literature incorporate spatial context information by using a fixed neighborhood window around a pixel or by relying on segmentation maps. However, by first reducing the hyperspectral data to only several PCA components, the spectral characteristics of the images are not used in a principal manner. Our proposed approach, by contrast, fully exploits the available spectral information in a hyperspectral image.

4.4 Proposed model

The goal of a majority of feature extraction methods for HSI classification is to reduce the dimensionality of the hyperspectral data while preserving as much of the discriminative information as possible, so that in a later stage a classifier can be trained on the extracted features. Since it is difficult to discern potentially relevant features from hyperspectral data, we approach hyperspectral image classification as an end-to-end learning [94, 95] task, where the assignment of class labels from hyperspectral input pixels is a single stage learning process, in which the intermediate feature representations are also learned. Therefore, we propose a CNN model for HSI classification. Additionaly, we investigate hyperspectral data augmentation as a way of mitigating the problem of limited training samples in hyperspectral image classification.

Deep CNNs have been successfully applied in solving challenging tasks, such as image classification [71], speech recognition [96], music information retrieval [97], and text recognition [95]. However, due to network generalization issues [98], deep CNNs for image classification tasks require a large number of images to prevent overfitting, and thus

appear inadequate for the HSI classification problem, where a dataset typically consists of a single capture of a scene. Furthermore, the large number of bands in hyperspectral images pose a computational challenge for a straightforward application of a CNN model.

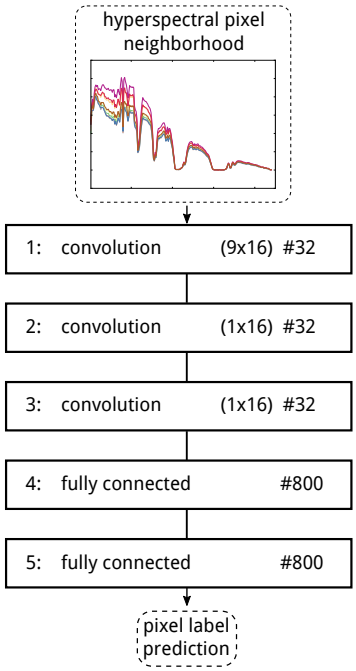


Figure 4.5: Diagram of the proposed convolutional neural network architecture for hyperspectral image classification. The size of the filters in the convolutional layers are indicated as $(h \times w)$, and # denotes the number of convolutional filters, as well as the number of hidden units in the fully connected layers.

We propose a CNN architecture which integrates both spatial and spectral information for simultaneous spatial-spectral classification of hyperspectral images. The proposed architecture is visualized in Figure 4.5. The input to the network consists of the eight-connected neighborhood of a hyperspectral pixel, to account for the spatial information context. In order to exploit the original spectral information, all convolutional operations are performed across the spectral bands. The network consists of 5 layers: three convolutional layers with width 16, followed by two fully connected layers with 800 units each. Note that the size of the filters in the first convolutional layer

is 9×16, where the first dimension accounts for the total number of pixels in the spatial neighborhood window of the input pixel, and the second dimension is the width of the filter. This allows for simultaneous learning from both the spatial and spectral domain.

In order to obtain the CNN architecture from Figure 4.5, we experimented with the number of layers, the number of hidden units in the fully connected layers, and the number and size of the filters in the convolutional layers. In addition, we tested several modifications of the original network. Namely, we experimented with max-pooling layers after the convolutional layers, and also with varying the stride of the convolutions. This worsened the classification results, which is indicative of non-stationarity of statistics across spectral bands. Testing the hyperbolic tangent activation function produced slightly better results than rectified linear units [99] activation. As a result, we used hyperbolic tangent activations in all layers, with the exception of the last layer, where the softmax function was used. We also attempted dropout regularization [100] in the fully connected layers. However, this did not improve the classification results.

We trained the network using minibatch gradient descent and momentum [101], and we set the size of the minibatches to 50 samples. Backpropagation was used to obtain the gradients of the error, placing the computational complexity of model training to $O(W^2)$, where W denotes the number of adaptive parameters. We evaluated the model on a held out validation set during training, and we report results on a separate test set for the model that achieved the best results on the validation set.

4.5 Data augmentation

Identifying the classes of pixels from hyperspectral images to produce labeled training data is a manual task, which is expensive and time consuming. Therefore, available training samples for HSI classification are scarce. To try to alleviate this problem, we experimented with simple augmentation for hyperspectral data. For each class in the hyperspectral image dataset, we calculate the per-spectral-band standard deviation of the samples in the training set which belong to the class. Afterwards, we use the calculated vector of standard deviations σ as a parameter to a zero mean multivariate normal distribution $\mathcal{N}(0, \alpha\Sigma)$, where α is a scale factor, and Σ is a diagonal

matrix containing σ along the main diagonal entries. Finally, the augmented samples for the class are generated by adding noise sampled from the distribution $\mathcal{N}$ to the original samples. We tried several values for the scaling factor in the set $\{1, 0.5, 0.33, 0.25, 0.125\}$, and fixed $\alpha = 0.25$ for the experiments. The goal of the proposed hyperspectral data augmentation is to prevent overfitting in cases where a low number of samples are used to train the network.

4.6 Experimental results

We tested our method on the commonly-used Indian Pines hyperspectral image dataset [3]. This dataset was acquired in June 1992 by NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). The Indian Pines scene is a mixed forest and agricultural site in Northwest Indiana, captured at about 20 km altitude by the AVIRIS sensor. The hyperspectral image of the scene consists of 220 bands in the spectral range from 0.4 μm to 2.5 μm, with a spectral resolution of 10 nm. The whole scene consists of 145 $\times$ 145 pixels. There are in total 10 366 labeled samples. With a moderate geometrical resolution of 20 m per pixel, and 16 land cover classes, this dataset poses a challenging classification problem due to the unbalanced number of samples per class, and high inter-class similarity of samples in the dataset.

Table 4.1: AVIRIS Indian Pines dataset and per class training sets and corresponding test sets.

Class		Train set			Test set		
#	Name	5%	10%	20%	5%	10%	20%
1	Alfalfa	3	6	11	25	24	21
2	Corn-notil	72	144	287	681	645	573
3	Corn-min	42	84	167	396	375	333
4	Corn	12	24	47	111	105	93
5	Grass-pasture	25	50	100	236	223	198
6	Grass-trees	38	75	150	354	336	298
7	Grass-pasture-mowed	2	3	6	12	11	10
8	Hay-windrowed	25	49	98	232	220	195
9	Oats	1	2	4	9	9	8
10	Soybeans-notil	49	97	194	459	435	387
11	Soybeans-min	124	247	494	1 172	1 110	987
12	Soybeans-clean	31	62	123	291	276	245
13	Wheat	11	22	43	100	95	84
14	Woods	65	130	259	614	582	517
15	Bldg-grass-trees	19	38	76	180	171	152
16	Stone-steel-towers	5	10	19	45	42	38
	Total	524	1 043	2 078	4 917	4 659	4 139

For our experiments, we evaluated the classification accuracy of

the method using a balanced training set per class, with low number of training samples. We trained the network with 5%, 10%, and 20% of randomly selected labeled samples per class, and equally divided the remaining labeled samples into separate validation and test sets. In each case, we repeated the experiment with and without hyperspectral data augmentation.

Table 4.2: Classification results for the Indian Pines image on the test sets.

Indian Pines		Test set		
		5%	10%	20%
Augmented	OCA(%)	86.54 ± 0.30	92.70 ± 1.00	96.58 ± 0.55
	F1 score	0.86 ± 0.00	0.93 ± 0.01	0.97 ± 0.01
Non-augment.	OCA (%)	85.46 ± 1.73	92.76 ± 0.93	96.54 ± 0.47
	F1 score	0.85 ± 0.02	0.93 ± 0.01	0.96 ± 0.00

The achieved classification results for each of the experiments are shown in Table 4.2. We performed 5 Monte Carlo runs, where for each run we selected a training set of 5%, 10%, and 20% of the labeled samples, as explained above, to train our model. In the cases with augmentation, we found 3 fold (per class) augmentation of the training data to give the best results. We report the average and standard

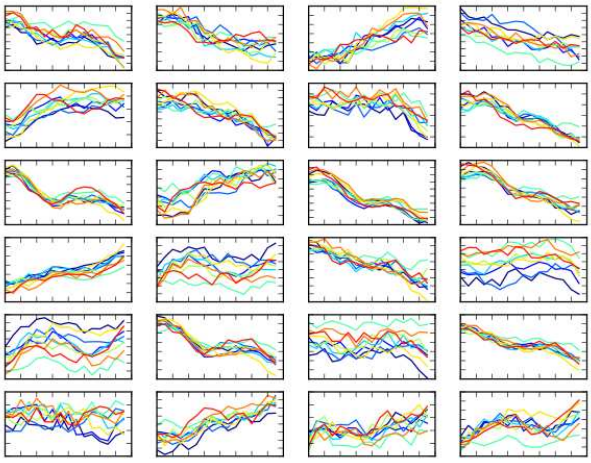


Figure 4.6: A subset of filters learned in the first convolutional layer of the network. Each subplot represents a (9×16) filter.

error of the 5 Monte Carlo runs in terms of the overall classification accuracy (OCA), i.e., the number of correctly classified samples from the total number of samples in the test set, and the F1 score, which

is weighted so that it accounts for the imbalance of the classes. From the results in Table 4.2, it can be seen that only when using a very low number of augmented labeled samples for training (5%), there is improvement in the classification scores over the non-augmented counterpart. However, we have observed that in all cases, augmentation reduced the number of training iterations significantly, as compared to training with the corresponding non-augmented data.

We have visualized some of the learned filters from the first convolutional layer of the network in Figure 4.6. From the visualization, it is clear that the learned filters have a structured shape, and that some of the filters roughly resemble different spectral band-pass filters.

4.7 Conclusions and original contributions

Due to the inherent nature of hyperspectral data, discernment of good features for hyperspectral image classification is difficult. Therefore, in this chapter, we have presented a new approach towards hyperspectral image classification based on deep convolutional neural networks. To evaluate the effectiveness of the method, we performed experiments on a commonly-used hyperspectral image dataset. Our experimental results have shown that the neural network model can learn structured features resembling different spectral band-pass filters directly from the input data. These features prove useful for hyperspectral image classification, which makes end-to-end learning applicable to hyperspectral scene understanding.

The work presented in this chapter is published in *Proceedings of ACM International Conference on Multimedia 2015*:

- Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. Hyperspectral image classification with convolutional neural networks. *ACM MM'15*.

Chapter 5

Fault Detection for Rotating Machinery

Vibration analysis is a well-established technique for condition monitoring of rotating machines, as the vibration patterns differ depending on the fault or machine condition. In this chapter, we propose two systems for bearing fault detection, one based on engineered features, and another feature learning system based on the model developed in the previous chapter. Experimental results on a dataset including several fault conditions show that our feature learning system can outperform traditional systems using hand-crafted features, while relying only on limited knowledge about vibration analysis.

5.1 Introduction

Condition monitoring (CM) is used to inspect the state of machines and to detect faulty components. It is crucial in reducing operational costs, prolonging the lifetime of machines, and enhancing operational uptime. Components that are often the main source of failure in rotating machines, such as wind turbines, are rolling-element bearings [102]. To monitor the condition of machine components, such as rotors, shafts, couplings, gears, and also bearings, vibration analysis is often used. The presence of the rolling elements in the bearings induce vibrations that are inherent to the system. The position of the rolling elements change continuously with respect to the load, causing a behavior that depends upon the rotation speed. Furthermore, geometrical imperfections or surface roughness also cause vibrations. Not

only are vibrations generated in normal operational conditions, but also due to faults, such as outer-raceway faults, inner-raceway faults, rolling-element faults, cage faults, imbalance, and misalignments.

To detect if a fault is present, a frequency spectrum analysis is often done [103]. This technique requires the frequency spectrum to be calculated together with the fundamental frequencies of the bearings. The amplitude at these frequencies can then be monitored for anomalies. However, such a technique has many disadvantages. First, the frequency calculations have the assumption that there is no sliding, i.e., the rolling elements only roll on the raceways. Nevertheless, this is seldom the case. Often, a bearing undergoes a combination of rolling and sliding. As a consequence, the calculated frequencies may differ slightly compared to the actual frequencies. Second, if multiple faults occur simultaneously, the frequencies generated can add and subtract, obfuscating important frequencies. Third, there is also the possibility that interference is induced due to other vibrating components, hence obscuring useful features. Lastly, some faults, such as smearing faults, do not even manifest themselves as a new cyclic frequency [104], which makes them very hard to detected via traditional vibration analysis techniques. Because of these various challenges, analysis of frequency spectra of vibration signals can be difficult to interpret, especially in a real-time manner, other than by an experienced vibration analyst [103].

Vibration patterns differ depending on the fault condition of the machine, which makes vibration analysis a suitable technique for condition monitoring of rotating machines. However, mainly engineered features are currently used for automatic fault detection. Unfortunately, these require human expert knowledge for designing the features and possibly interpreting the results. In Chapter 4, we presented a new method for classification of hyperspectral images based on convolutional neural networks. In this chapter, we will apply a similar CNN-based learning model to the one developed in Chapter 4, but for vibration analysis of different bearing fault conditions which occur in rotating machines. We also develop a system for classifying bearing faults based on different engineered features, and compare the classification results of the two approaches on a dataset containing samples of several fault conditions.

The remainder of this chapter is organized as follows. An overview of related work on early fault detection for condition monitoring of

rotating machines is given in Section 5.2. In Section 5.3, we present the test set-up used in the creation of a bearing fault dataset for our experiments. Section 5.4 is devoted to the proposed fault classification methods based on engineered features and learned features. The results achieved by both approaches are evaluated and compared in Section 5.5. Finally, we conclude this chapter with Section 5.6.

5.2 Related work

Different specific fault conditions can be detected from the machine's vibration patterns. One example is imbalance, which is caused due to the shift between the principal axis of inertia and the axis of rotation, and results in a high amplitude at the rotation frequency of the machine in the frequency spectrum [105]. Other faults which can be detected in a similar manner are damaged raceways, since these faults generate a peak at a specific fundamental frequency [106]. Besides indicative frequency features, it has also been shown that certain time-based statistical features, such as kurtosis and crest, are useful in identifying a defect bearing [107]. Furthermore, the root-mean-square (RMS), another time-based feature of the vibration signal, is indicative of the amount of separation between the rolling elements and the raceways due to lubrication in a linear bearing [108]. Although several different features can be extracted from vibration data, identifying the different machine conditions and anomalies is still a daunting task for a human expert. Therefore, as described below, there have been efforts to automate the interpretation process.

Learning algorithms for machine fault detection focus on two aspects of the problem: anomaly detection, and condition, or fault classification. Anomaly detection is the process of identifying measurements that do not conform to the general patterns of the dataset [109]. Here it is assumed that the deviations of the machine's normal operating state, the anomalous measurements, correspond to a machine fault. Anomaly detection algorithms do not require samples from all the different possible fault conditions for training, but only samples taken during normal operating conditions and any abnormal measurement samples. Therefore, anomaly detection datasets consist of only positive and negative samples. In the context of anomaly detection, the different vibration features discussed previously are used by algorithms such as one-class SVMs, Gaussian distribution fitting,

clustering in combination with PCA, hidden Markov models (HMMs) and neural networks [109–112].

Contrary to anomaly detection, condition or fault classification methods require data from the different fault conditions, in addition to the measurements from the normal operating mode of the machine. The advantage of these methods to anomaly detection methods, however, is that they can be used to identify which specific fault has occured. Fault classification methods make use of the same kind of features as discussed above, in combination with classifiers, such as k-nearest neighbor, naive Bayes, decision trees, and NNs [113–115]. Using such fault classification approaches, several types of faults can be accurately identified, such as inner-raceway faults, outer-raceway faults, and rolling element faults. However, some faults are more difficult to identify reliably, such as lubricant starvation [104], which can be caused due to grease dry-out.

Lubrication controls many properties of a system, such as friction, wear, contamination, temperature levels, and corrosion. Lack of lubricant is often the root cause of many bearing failures [116]. If lubricant starvation is not detected in time, other additional faults may be induced, making it more difficult to identify every individual fault [115]. Therefore, more advanced techniques are required when several faults are present in a rotating system at the same time. As a result, methods that try to learn useful features for machine fault classification from vibration data have been recently proposed in the literature. In the method proposed by Wang et al. [117], sparse modeling is used to learn dictionaries out of sub-bands obtained by decomposing the vibration signal with a wavelet packet transform. Then, the residuals of the reconstructed signal, calculated by using each of the learned dictionaries, are used to form a feature vector. Often, due to vibrations of other mechanical components, vibration signals contain more than just the vibrations generated by a monitored component, which makes the identification of a fault more difficult. In the work of Deng et al. [118], sparse coding and online dictionary learning is used to denoise signals generated by an aircraft engine, and extract impulse features, which allows a fault frequency to be accurately identified. The sparse coefficients used in their approach are obtained by fusion of sparse coefficients estimated with several sparse coding algorithms and the learned dictionary. Neural network models have also been used for learning features from vibration data. In particu-

lar, stacked sparse autoencoders have been applied for machine fault detection [119]. AEs [72] try to learn the inherent structure of the underlying distribution of the data by learning the identity function from unlabeled training samples, but with a constrained number of hidden units. Sparse AEs impose a further constraint on the activations of the hidden units through the model's objective function, thus forcing the output of the hidden units to be sparse, which results in learning relevant features from the data.

5.3 Bearing fault dataset

In order to evaluate the proposed fault classification approaches using different types of faults, we created a bearing fault dataset using the setup depicted in Figure 5.1. The technical specifications of the different parts of this setup are summarized in Table 5.1.

There are two bearing housings in the setup. Out of the two housings, the housing farther from the motor contained the different fault induced bearings during the CM runs. On this housing, two accelerometers were mounted perpendicular to one another in order to measure the vibrations on top of the housing, and on the back of the housing. The various fault conditions introduced are the following:

1. Healthy bearing (HB).
2. Mildly inadequately lubricated bearing (MILB).
3. Extremely inadequately lubricated bearing (EILB).
4. Outer-raceway fault (ORF).
5. Healthy bearing during imbalance (HB-IM).
6. Mildly inadequately lubricated bearing during imbalance (MILB-IM).
7. Extremely inadequately lubricated bearing during imbalance (EILB-IM).
8. Outer-raceway fault during imbalance (ORF-IM).

Some images of the induced conditions are presented in Figure 5.2. To imitate an ORF, three small shallow grooves were added mechanically on the bearing's outer-raceway (Figure 5.2c). Also, grease was

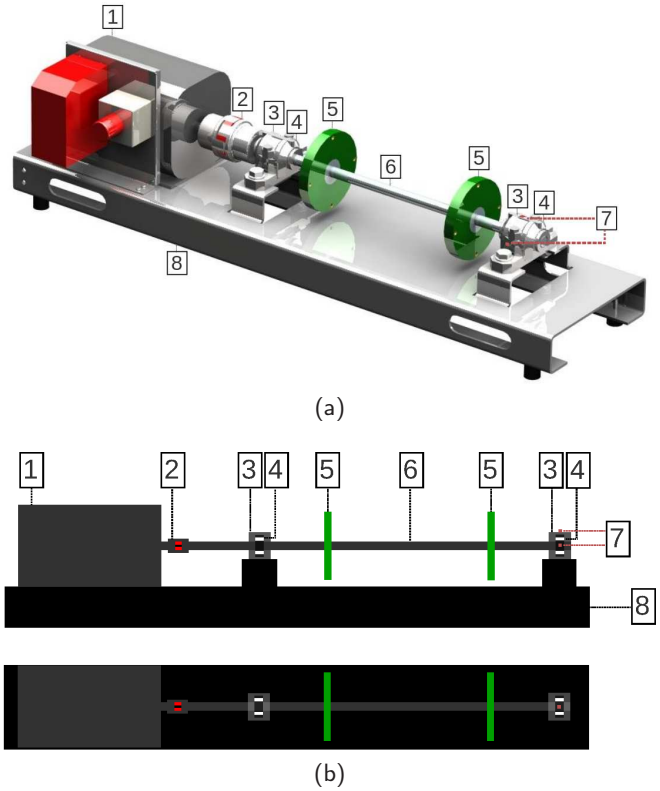
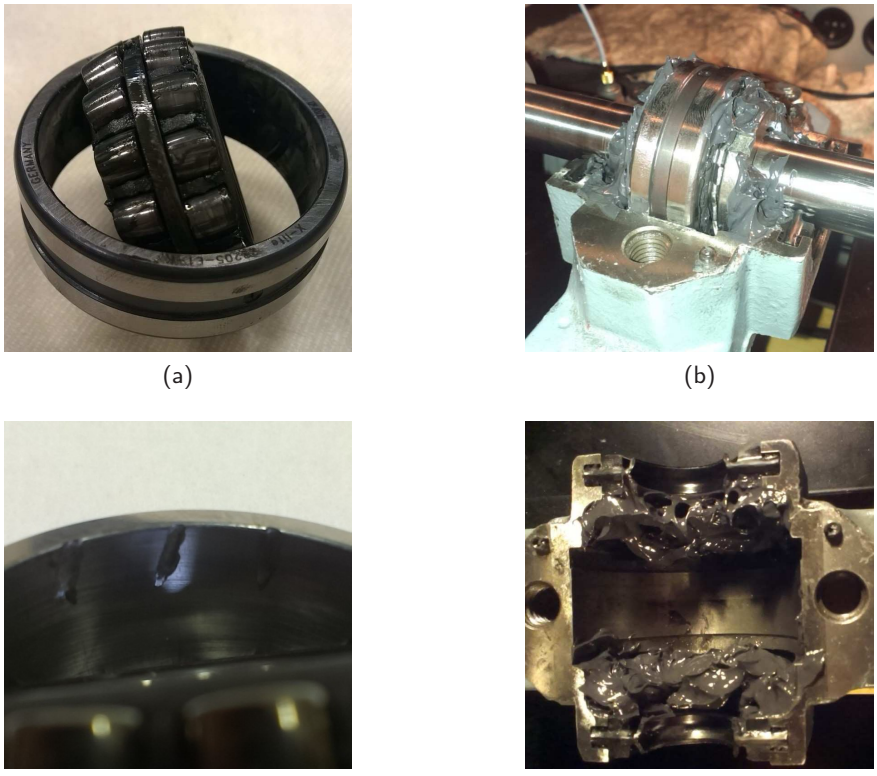


Figure 5.1: Setup used in the creation of the bearing fault dataset. (5.1a Three-dimensional model. (5.1b) Side and top views. The setup consists of the following components: servo-motor (1); coupling (2); bearing housing (3); bearing (4); disk (5); shaft (6); accelerometer (7); metal plate (8).

Table 5.1: Technical specifications of the setup components.

Property	Value
Bearing code	FAG 22205-E1-K
Bearing type	Spherical roller bearing with tapered bore and adapter sleeve
Housing code	SNV052-F-L
Housing type	Closed plummer block
Grease	Molykote BR 2 plus
Rotation speed	25 Hz
Accelerometer type	IEPA 4534-B
Accelerometer frequency	51 200 Hz



(a) (b)

(c) (d)

Figure 5.2: Examples of different bearing conditions and the grease reservoir. Mildly inadequately lubricated bearing outside the housing (a). Healthy bearing in an open housing (b). Three small, shallow grooves to imitate an outer-raceway fault (c). Grease reservoir in the bottom of the housing (d).

added to the bearing as lubricant. The amount of required grease was calculated as $D \times B \times 0.0027$ [g], where D is the outer diameter of the bearing, and B the inner diameter. The bearings used in the setup runs had an outer diameter of 52 mm and an inner diameter of 18 mm. Both the HBs and those with an ORF contained 2.5 g of grease, in addition to the 20 g of grease in the grease reservoir within the housing. The amount of grease was determined so that the housing cavities are filled to the recommended 60% [120]. For the MILBs, the grease reservoir is removed and the grease on the bearing is diluted. Similarly, for the EILBs no reservoir is present, and the grease in the bearings is diluted further. All four conditions were also subjected to rotor imbalance. The imbalance was created by adding a 13 g bolt to the outer disk at a radius of 5.4 cm. By means of this setup, a data set was created incorporating the eight different bearing conditions. For every condition, five bearings were used, resulting in 40 test runs in total. Each test had a runtime of one hour, from which the last 10 minutes of vibration data were captured using the accelerometers. In the next section, the proposed techniques for identification of the described fault condition are discussed in detail.

5.4 Fault classification

The goal of the proposed approaches is to detect fault conditions, and to classify the type of fault that occured. For the feature engineering and feature learning method alike, we divide fault classification in two classification tasks: one for the machine condition and one for the bearing condition. It has been determined experimentally [7] that a two pipeline system is advantageous to a single pipeline system for fault classification. Therefore, we regard the fault detection or classification as a combination of a binary classification problem and a multi-class classification problem. Every 10-minute vibration recording is classified by the binary classifier as balanced or imbalanced, and by the multi-class classifier according to HB, MILB, EILB, or ORF. By using a system with two pipelines, the combination of the two labels generated for each sample give the final fault condition (one of the eight classes as listed in Section 5.3).

5.4.1 Feature engineering method

The general architecture of the proposed feature engineering method for bearing fault classification is given in Figure 5.3. The goal of pipeline one is to determine if there is rotor imbalance. The second pipeline deals with identifying the specific bearing fault (HB, MILB, EILB, or ORF).

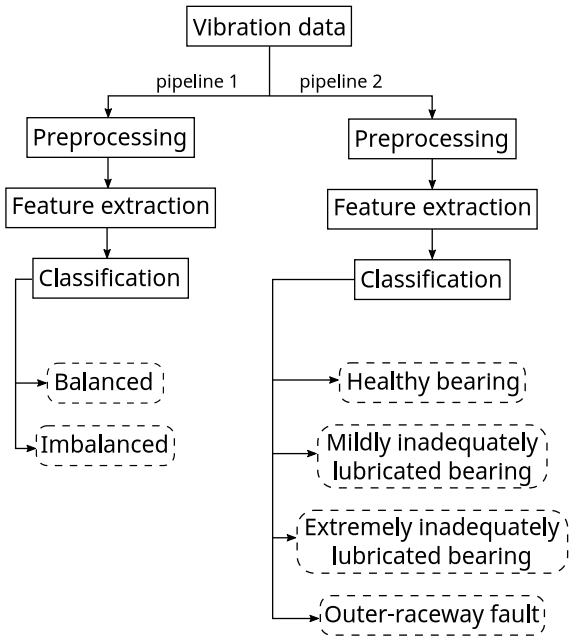


Figure 5.3: Architecture of the system using engineered features for fault classification.

5.4.1.1 Pipeline one

As discussed in the related work in Section 5.2, imbalance can be detected by observing if there is a high amplitude at the rotation frequency of the machine. Note that the sampling frequency of the accelerometers is very high compared to the the rotation frequency of the machine, which is 25 Hz. However, to detect specific bearing faults in the second pipeline, the resonant frequency of the bearings needs to be measurable. This frequency is usually above 10 kHz, therefore, the sampling frequency has to be at least 20 kHz. Nevertheless, imbalance can also be detected using the chosen accelerometers. The first step

to extract the amplitude at the rotation frequency is windowing. A window contains one minute of vibration data, and overlaps by 50% with its neighboring window. This means that from every 10-minute vibration data recording, 19 windows are extracted, each containing 60 seconds $\times$ 51.2 kHz $=$ 3 072 000 samples. As there are two accelerometers mounted on the bearing housing, there are in fact twice as many samples. The window length is experimentally determined, and provides the most optimal results. Also, a relatively large window is preferred as it enables a small bin resolution for the discrete Fourier transform (DFT), which is used in the second step. In fact, when applying the DFT, the frequency resolution is $1/\textit{window length}$ or 0.0166 Hz/bin for a window length of 60 seconds, allowing for small frequency differences to be detected.

An example of a frequency plot is given in Figure 5.4. As can be seen from Figure 5.4, a peak close to the rotation frequency can be observed when there is imbalance. In the final step, the maximum frequency bellow 90 Hz is chosen as a feature from the frequency spectrum. This feature is extracted for the two vibration signals, per window, resulting in 19 samples per test run, each containing two features. After the features are calculated, classification is applied. A

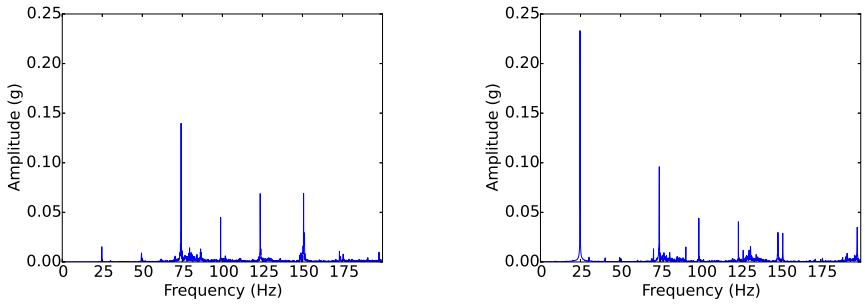


Figure 5.4: Plots of the frequency spectrum of bearings during normal operating conditions (a), and when there is imbalance (b). A dominant peak at approximately 25 Hz can be clearly seen in (b).

plot of the features extracted from the measurements during imbalance and normal operating conditions provide effective discrimination between the two different samples with a linear discrimination boundary (Figure 5.5). Therefore, a simple classification method suffices to

classify the samples. Logistic regression was chosen in this case, although, other options such as a linear support vector machine or a decision tree could be equally valid.

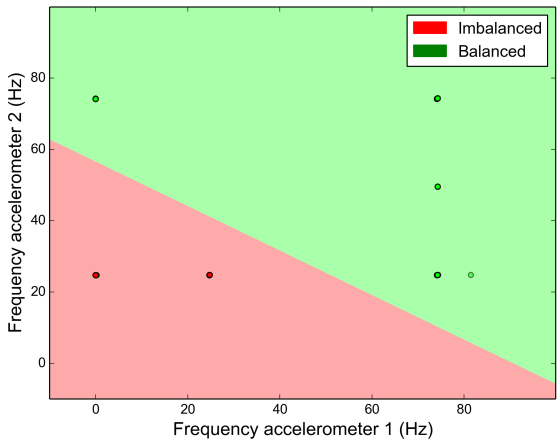


Figure 5.5: A scatter plot of the maximum frequencies from both accelerometers for all the samples. The decision boundary determined by logistic regression is also plotted. Accelerometer 1 is located on top of the housing, and accelerometer 2 on the side of the housing.

5.4.1.2 Pipeline two

Identifying the specific bearing condition, which is the task of the second pipeline, is a more difficult problem than the binary classification of the machine condition as balanced or imbalanced. Therefore, a larger set of features is used in this pipeline. Similar to the first pipeline, windowing is applied in this pipeline too. From every window, several features are calculated. First of all, three statistical features are calculated: the RMS, kurtosis, and crest factor. These features have proven useful for bearing fault detection [107,108]. The RMS, kurtosis, and crest factor are calculated according to Equations (5.1), (5.2), and (5.3) respectively, where $\mathbf{x}$ is a vector of N samples in a window, and μ and σ respectively denote the mean and

the standard deviation of $\mathbf{x}$.

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i}^{N} x_i^2}. \tag{5.1}$$

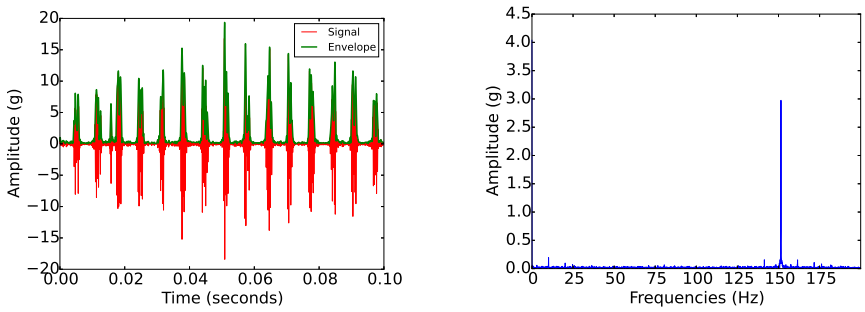
$$\text{Kurtosis} = \frac{\sum_i^N (x_i - \mu)^4}{N\sigma^4}. \tag{5.2}$$

$$\text{Crest} = \frac{\max(|\mathbf{x}|)}{\text{RMS}}. \tag{5.3}$$

When a rolling element hits a fault in the outer raceway, the natural frequency of the raceway is excited, resulting in a high frequency burst of energy, which decays and is then excited again as the next rolling element hits the fault. This high frequency impulse is superimposed, that is, amplitude modulated on a carrier signal which originates from the rotating machine. To identify a fault, it is necessary to detect the frequency of occurrence of these high energy bursts. Therefore, envelope detection is applied. First, a band pass filter is used. All frequencies bellow 1 kHz, such as the carrier frequency, are removed. Also, frequencies above 20 kHz that interfere with high frequency signals originated from the impact are filtered out. After this filter process, the high frequency impacts should be better isolated. The final step is to determine the envelope signal which will have a frequency equal to the frequency of occurrence of the high energy bursts. The envelope is determined by taking the magnitude of the analytical signal, which is computed using the Hilbert-Huang transform. An example of this envelope signal can be seen in Figure 5.6. When there is an outer-raceway fault, the frequency of the envelope signal will manifest itself at the ball pass frequency of the outer raceway (BPFO).

The BPFO can be calculated using Equation (5.4), where n is the amount of rolling elements, f the rotation frequency, d the diameter of the rolling elements, D the diameter of the rolling element cage, and α the contact angle. This results in a BPFO at 150.41 Hz for the chosen bearings. To summarize, if the frequency of the envelope signal is near the BPFO and has a high amplitude, it can be concluded that an outer-raceway fault is present. An example of this can be seen in Figure 5.6b. From the envelope frequency, the maximum amplitude, and the corresponding frequency are extracted as features. These two features are calculated for both vibrations signals.

$$\text{BPFO} = \frac{1}{2} n f (1 - \frac{d}{D}\cos\alpha) \tag{5.4}$$



(a) (b)

Figure 5.6: Vibration signal generated by an outer-raceway fault and its frequency spectrum. The signal together with the corresponding envelope (a). Frequency spectrum of the envelope signal (b).

As in pipeline one, all features are extracted from overlapping windows on the vibration signals, resulting in 19 samples per measurement. Each sample consists of 14 features (seven features per accelerometer): RMS, kurtosis, crest factor, frequency of the highest amplitude of the envelope signal spectrum, the maximum amplitude in the envelope signal spectrum, rotation frequency, and amplitude of the rotation frequency. The rotation frequency and its amplitude were also included as they improved the classification results.

Discriminating between the four different faults is a more difficult task, hence a random forest classifier is chosen [121]. A RF classifier is a non-linear, multi-class ensemble classifier based on decision trees. Due to this ensemble technique, parallelism is inherent to a RF, enabling a fast training phase. Also, a RF requires a minimal amount of meta parameters to tune. The most important parameter is the number of individual decision trees contained in the forest, which we fix at 200 trees, since adding more trees did not improve the results further. For comparison purposes, tests were also done using a SVM with different kernels: a linear kernel, a polynomial kernel, and a RBF kernel. For the SVM, the hyper parameters C, which determines the penalty on misclassifications, γ, which determines how far the influence of a single training example reaches, and the degree of the polynomial kernel, were determined using grid search optimization.

5.4.2 Feature learning

Similar to the feature engineering based approach, the feature learning based approach uses a two pipeline system as depicted in Figure 5.7. Since the binary classification problem of balanced versus imbalanced samples can already be effectively solved using pipeline one of the feature engineering approach, we will reuse this pipeline here. Nevertheless, for the detection of the four specific bearing conditions (HB, MILB, EILB, and ORF), a feature learning model is proposed, which forms the second pipeline.

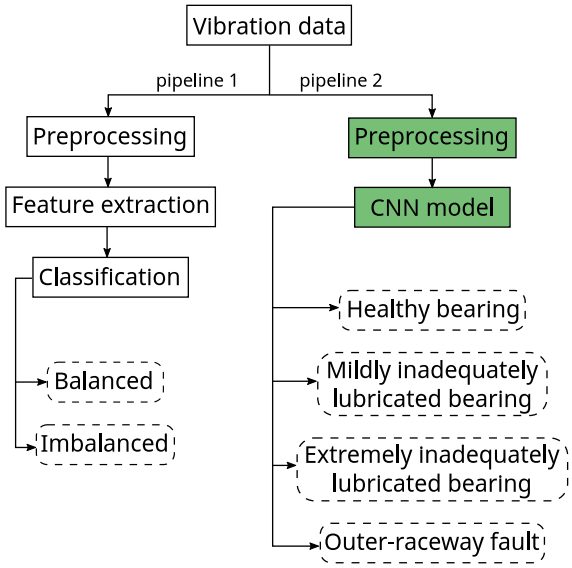


Figure 5.7: High-level representation of the proposed feature learning architecture. The modules in green indicate the modified parts compared to the feature engineering based approach.

5.4.2.1 CNN model

Our proposed feature learning approach is based on a convolutional neural network model. More specifically, a CNN model similar to the one proposed in Chapter 4 is used. However, the model applied here leverages the capacity of the network for exploiting the spatial structure in data to effectively capture the covariance of the frequency decomposition of the accelerometer signals. Note that the two accelerometers are placed perpendicular to one another, and the goal

here is to differentiate between the complex bearing conditions by learning the patterns of changes of the joint accelerometer signals. Figure 5.8 shows a diagram of the proposed CNN architecture.

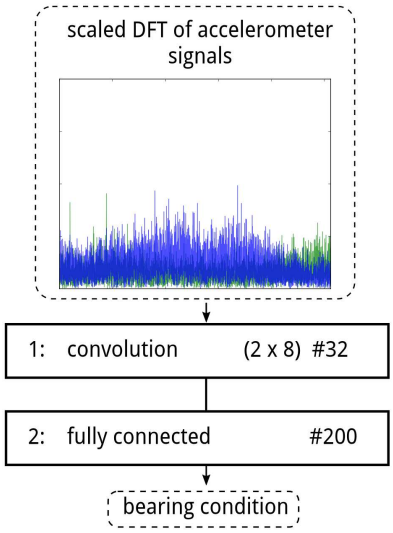


Figure 5.8: Architecture of the CNN model for bearing fault detection. The size of the convolutional filter is denoted as $(h \times w)$, and # denotes the number of convolutional filters, and the number of hidden units in the fully connected layer.

The architecture which yielded the best results in the experiments consists of one convolutional layer with width 8, followed by a fully connected layer with 200 units. The height of the convolutional layer corresponds to the two signals originating from the accelerometers. The input signals are preprocessed in order to train the model. First, the accelerometer signals are scaled to have zero mean and unit variance. Then, from the training set signals, non-overlapping windows are extracted containing one second of measurement samples. For each window of extracted samples, the DFT is calculated. The amplitudes of the frequency decompositions are then used as training samples for the neural network model. It was determined that the accelerometer's sampling resolution could be lowered without affecting the output of the model. Therefore, 5-fold subsampling is applied on the original accelerometer data. The CNN model was trained using minibatch gradient descent and momentum [101], using 100 training examples per minibatch.

It has been shown that by using a deep architecture, i.e., a network with many layers, the network becomes more robust to the variation in the data [122]. Hence, if the dataset has a lot of variation, a deep architecture is required. As the manifestation of the different faults considered here shows little variation, a shallow architecture suffices. Furthermore, the initial layers of CNNs learn the fastest, therefore a short training time is sufficient to achieve convergence [122]. Several variations of the proposed network were tested by varying the number of convolutional and fully connected layers, and the number of units per layer. For our particular use case, it was determined that a deep version of the proposed architecture does not yield better results.

5.5 Experimental results

In this section, we evaluate the performance of the proposed feature engineering and feature learning based approaches, and discuss the obtained results. The following methodology was used in conducting the experiments. To quantify the performance of the different classifiers, four error measurements are calculated: accuracy, precision, recall, and F1 score. The four different metrics are chosen because they directly reflect the impact on CM requirements. If a CM system triggers an alarm when the classifier supposedly detects a fault, it is more interesting to be alerted of all the detected faults, even if there are some false alarms. However, the operator does not want to have too many false alarms, since this increases the operational cost due to unnecessary downtime. In other words, if many alarms are triggered, a lot of faults are brought to the operator's attention (higher recall), nevertheless, there are also more false alarms (lower precision). On the other hand, if only real faults are flagged, but some faults are missed, and there are no false alarms, there will be a high precision and a low recall. A good classifier will maximize both, so that an alarm is only triggered when there is an actual fault, with low number of missed faults and false alarms. This combination is expressed directly in the F1 score. The accuracy is also chosen, because it provides a general measure of the total amount of correctly classified samples.

To evaluate the performance of the systems under a real-world scenario, we employed leave-one-out cross-validation on bearing level. That is, from the 40 recordings, 32 recordings (corresponding to four

different bearings) are used to train the system, and eight recordings (all corresponding to the different fault conditions of the fifth bearing) are used for testing. This procedure is repeated five times, so that each of the five bearings can be used for testing, which ensures that the results are not biased towards a specific run.

5.5.1 Feature engineering results

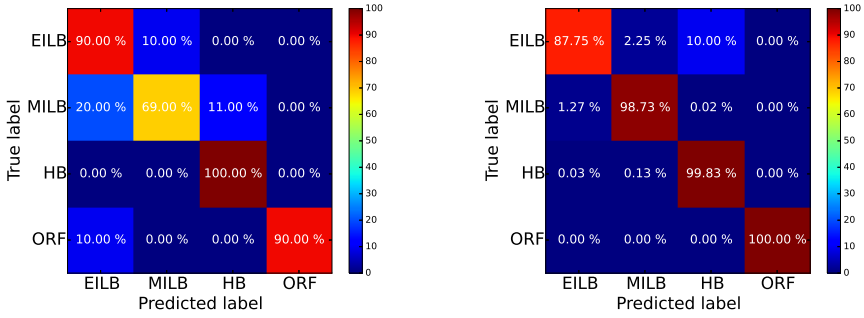
As illustrated in Figure 5.5, distinguishing between a balanced and imbalanced system can be solved with absolute accuracy. That is, the mean accuracy, recall, precision and F1 score, achieved by the logistic regression during cross-validation, are 100 % ($\sigma = 0\%$). Since only the rotation frequencies measured by the two accelerometers are used as features, it is possible to see which of the accelerometers provides more discriminative information. Therefore, we repeated the classification task using a random forest classifier. For the rotation frequency extracted from the accelerometer on top of the housing, the importance of the feature is 73.28% ($\sigma = 5.90\%$), and for the accelerometer on the side of the bearing housing 26.72% ($\sigma = 5.90\%$).

The results achieved by the second pipeline are summarized in Table 5.2. As can be seen, the RF classifier outperforms the different SVMs. In general, the classification of the different bearing fault conditions is more difficult. Inspecting the confusion matrix of the RF classifier (Figure 5.9a), it can be seen that the system can identify a healthy bearing perfectly. Also, outer-raceway faults are EILBs have a high classification accuracy. On the other hand, the class of MILBs is the most difficult to detect, as it can be confused with an EILB or HB. This is possibly due to a lack of discriminative information in the engineered features for this class, which hinders the disambiguation of MILB samples from the similar EILB samples, and to a lesser extent from the HB samples as well.

For the second pipeline, the importance of the features can also be obtained from the RF classifier (see Table 5.3). Several observations can be made from these statistics. First, the amplitude of the fault frequencies are important to the model, which is due to the direct relation to the outer-raceway fault. Second, the RMS metric is very important to the model, which is expected, as the RMS is possibly indicative of the separation (caused by the lubricant) between the rolling elements and raceways (as discussed Section 5.2 for a linear bearing). Third, the crest feature seems to be less important. A

Table 5.2: Performance results (%) of the system using engineered features for bearing fault classification. The average and standard deviation over 10 runs are shown.

Metric	RF	SVM (linear)	SVM (polynomial)	SVM (RBF)
Accuracy	87.25 ± 8.10	72.50 ± 18.37	80.00 ± 20.31	77.50 ± 18.37
Precision	89.83 ± 8.21	73.75 ± 19.70	80.00 ± 23.78	82.08 ± 15.78
Recall	87.25 ± 8.10	72.50 ± 18.37	80.00 ± 20.31	77.50 ± 18.37
F1 score	86.73 ± 8.14	73.12 ± 19.01	80.00 ± 21.91	79.73 ± 16.98



(a) (b)

Figure 5.9: Confusion matrices of the bearing faults classification using engineered features (a) and learned features (b).

likely explanation is that this feature incorporates the RMS, which is already directly available to the model, thus providing little additional information. Finally, the kurtosis feature is also less important to the model. Since kurtosis is indicative of bearing damage [106], possibly some of the required information are already directly provided to the model by the fault frequencies and the rotation frequencies.

Table 5.3: Mean and standard deviation (%) of importance scores obtained from the RF classifier for the engineered features used in the second pipeline.

Feature	Accelerometer (top)	Accelerometer (side)
RMS	18.49 ± 1.34	13.14 ± 1.48
Fault frequency amplitude	8.30 ± 0.80	14.88 ± 2.10
Fault frequency	4.86 ± 1.08	8.18 ± 0.70
Kurtosis	4.99 ± 1.24	3.48 ± 1.20
Crest	7.01 ± 0.89	1.44 ± 0.40
Rotation frequency	2.95 ± 0.39	3.07 ± 0.96
Rotation frequency amplitude	2.84 ± 0.37	6.37 ± 0.83

5.5.2 Feature learning results

As stated in Section 5.4.2, the feature learning based approach reuses pipeline one, which detects imbalance. For the second pipeline, which makes a distinction between HB, MILB, EILB, and ORF, the results are given in Table 5.4. As can be seen from the results, the performance is better for every metric. Based on the cross-validation results, we also conducted a paired two-tailed t-test. The results from the test confirmed that for every metric, the proposed CNN model performs significantly ($p < 0.05$) better compared to the feature engineering based approach. However, as can be seen in Figure 5.9b, the classifier still make some errors, as it classifies 10% of the EILB samples as HB, which leaves room for further improvement. The only class for which the classifier did not make any mistake is ORF.

The first pipeline, which is reused in both of the proposed systems, distinguishes between imbalanced and balanced samples, whereas the second pipeline distinguishes between HB, MILB, EILB, and ORF. The two pipelines operate independently from one another. In this case, the combined performance of a given system for the eight fault

Table 5.4: Performance results (%) of the CNN based system for bearing fault classification. The average and standard deviation over 10 runs are shown.

Metric	CNN model
Accuracy	91.77 ± 9.20
Precision	91.82 ± 9.23
Recall	91.76 ± 9.19
F1 score	91.79 ± 9.21

condition classes is the product of the performances of the two individual pipelines of the system. Here, in both the case of the system based on engineered features and the one based on learned features, the overall accuracy of the system is determined by the second pipeline because the first pipeline has an overall accuracy of 100%. Single pipeline systems using engineered as well as learned features were also tested, where the goal was to immediately classify between the eight fault conditions. The tests, however, indicated that the performance of such single pipeline systems was significantly lower when compared to the performance of the corresponding two pipeline system.

5.6 Conclusions and original contributions

In this chapter, we have presented two approaches for bearing fault classification, which is significant in condition monitoring of rotating machines. The proposed feature learning method based on CNNs has shown that better results can be achieved than with more standard systems, which rely on classical machine learning techniques in combination with hand-crafted features. This is particularly true for the detection of outer-raceway faults, different levels of lubricant degradation, and healthy bearings. For detecting imbalances, however, a small number of simple and well crafted features can prove sufficient for the task. The major advantage of feature learning techniques is that little domain expertise is required to achieve very good results. Nevertheless, a small number of classification errors still occur. An interesting venue of further research would be to test the feature learning model proposed here on more fault conditions. Furthermore, to decrease errors even more, additional complementary sensors could be considered. An example of a useful sensor for condition monitoring

is a thermal camera. It has been shown that by the use of thermal cameras lubrication degradation can be easily detected [7]. As such, the combination of thermal and vibration data can possibly be used in designing a strong multi-modal fault detection system.

The work presented in this chapter is submitted for publication in *Journal of Sound and Vibration*:

- Olivier Janssens[1], Viktor Slavkovikj[1], Bram Vervisch, Steven Verstockt, Kurt Stockman, Mia Loccufier, Rik Van de Walle, and Sofie Van Hoecke. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *Journal of Sound and Vibration (accepted for publication).*

[1]Contributed equally to this work.

Chapter 6

Conclusions

The capacity of machines for performing a number of intelligent tasks is preconditioned by their ability to correctly perceive objects from the environment and to detect the occurrence of events. Object classification and event detection are, therefore, two important problems in computer vision and artificial intelligence. In this dissertation, we have investigated instances of such problems, in particular, classification of terrains, fault detection in rotating machines, and detection of large-scale events such as wildfires.

For the terrain classification problem, we are interested in categorizing between different types of roads a vehicle is traversing along its route, or, for example, in identifying the various types of surfaces which can be found from an aerial or spatial view of an area. Accurate terrain classification is important for many applications. Classification of road surfaces, for example, is useful for navigation of autonomous vehicles, and automatic annotation of routes. For this purpose, we have proposed a multimodal road classification system in Chapter 2. A bicycle sensing setup was used to collect visual and vibration data along a route by utilizing a smartphone device mounted on the bicycle's frame. By engineering specific features which summarize important statistical characteristics of the road data modalities, and by training a robust nonlinear classifier on the calculated feature vectors, a high classification accuracy is achieved, thus enabling automatic annotation of recreational routes for users. Services, such as RouteYou[1], can use automatic route annotation methods to improve

[1] `http://www.routeyou.com`

their recommendations for recreational navigation.

Additionally, in Chapter 2, we have investigated the use of available online images for road classification, which removes any positioning requirements of the smartphone's camera that are necessary when capturing road surface images in our bicycle sensing setup. To test the applicability of such an approach, a comprehensive dataset of paved and unpaved road images has been created by employing user contributed route information (e.g., from OpenStreetMap[2]) and querying images from online services such as Google Street View. By learning features in an unsupervised manner, high classification accuracy was achieved on available online images of road surfaces without the need of domain knowledge to engineer features. Our comparative results show that such learned features provide on par classification results to those obtained by our feature engineered multimodal road classification system, when using only the system's visual modality features, and tuning them to perform well for the specific dataset.

From our road type classification experiments it can be concluded that, for datasets of limited size, feature engineering can be a powerful way of incorporating expert knowledge about the problem. However, the domain knowledge embedded in the feature design is task specific, which requires adjusting or even redesigning the features for similar problems and new datasets. Feature learning, on the other hand, does not necessitate prior knowledge about the task, and, given sufficient training examples, can provide just as good data representations, even for comprehensive datasets with high intra-class variability.

In the case of large-scale terrain classification, hyperspectral imaging, as discussed in Chapter 1, offers effective means for analyzing the surface characteristics of large areas. In this context, accurate terrain classification is a key component in several application areas such as environmental monitoring, urban planning[3], agriculture, and geology. However, the advantages of general hyperspectral image classification have also been widely adopted in fields such as biomedicine, information security, food sciences, and forensics. In Chapters 3 and 4, we investigate hyperspectral image classification in the framework of unsupervised and supervised learning, respectively.

Hyperspectral imaging draws its origins from spectroscopy—the study of radiation emitted or reflected from materials and its inten-

[2] `https://www.openstreetmap.org`

[3] `http://goo.gl/YY7FPD`

sity variation as a function of wavelength. Attributable to their origin, hyperspectral sensors are often referred to as imaging spectrometers. However, the identification of the unique spectra of materials in physical and analytical chemistry has also had an influence on methods for classification of images captured by hyperspectral sensors. Akin to classical spectroscopy, hyperspectral image classification methods have been developed which try to classify individual pixel spectra in terms of a mixture of several "pure" materials, or to use the spectral measurements in a holistic approach when training classifiers.

In Chapter 3, however, we have investigated the possibility of better exploiting the information in the correlated hyperspectral bands by learning features from subsets of the spectral domain. Although hyperspectral images contain a lot of information, the data is difficult to visualize, apart from a few bands in the visual modality. Therefore, the way to approach feature design from the visual and infrared modality is not straightforward. By unsupervised learning of basis representations from subsets of spectral bands, and by encoding convolutional samples from hyperspectral input pixels, an effective discrimination between land-cover classes could be achieved. Our results show that the proposed method compares favorably with other approaches that make use of unlabeled data for feature extraction. Furthermore, on several datasets, a classification accuracy was achieved surpassing or matching that of recent semi-supervised methods for hyperspectral image classification.

Building on the insights gained from the work described in Chapter 3, a novel supervised hyperspectral image classification approach is proposed in Chapter 4. By building a deep convolutional neural network model, an integral, end-to-end solution for hyperspectral image classification has been facilitated. From our experimental results, it can be concluded that the neural network model learns structured features resembling different spectral band-pass filters without prior knowledge. The structure of the learned filters are, in this case, intuitive to understand, however, the exact form of each individual filter, or the way of combining such band-pass filters is unlikely to be easily conceived by experts.

The classification method proposed in Chapter 4 is not solely applicable to hyperspectral images. Chapter 5 deals with classification of different kinds of faults that may occur in rotating machines, which is important for prolonging the machines' operational time [7,8]. With

a model similar to the one proposed in Chapter 4, but applied on vibration data, it is possible to learn the joint pattern of changes in the vibration signals, and thereby to distinguish between several types of faults with high accuracy.

Terrain classification from remote-sensing hyperspectral images of vast areas can also be used in the prevention, management, and response to large-scale hazard events, such as in the case of wildfires. Parallel to approaches that rely on information from specialized sensors (e.g., dedicated visual and thermal monitoring cameras), in Appendix A, we have investigated the use of social media information for wildfire detection and management. Through a review of methods, systems, and applications which make use of different modalities such as textual, visual, and geographic social media information, the advantages and drawbacks of using such information have been outlined for detection and management of wildfire hazards. The major drawbacks of using social media information arise from the noisiness of the data. In particular, relevant pieces of information from social feeds can be very sparse, and separating the important information from an extremely large volume of data can be challenging. The advantages, however, are multiple. Most importantly, information from social media can be used in addition, or complementary to that of traditional sources, and it is available at a very low cost. Furthermore, with the spread of mobile devices containing multiple sensors, human-centric sensing has become possible, where users are able to generate quantifiable sensory outputs. Distinctive to collective sensing is also the profusion of active social media users whose networks can span over large spatial areas. Finally, the immediacy and fast diffusion of news in such networks, and the specificity or locality of the propagated information, make social media particularly useful in cases of large-scale disaster events. Considering that current use of social media in the context of large-scale hazard events has only just started to gain traction, further research in this area holds important prospects for future systems for wildfire detection and management.

To conclude, the use of multimodal information has been a recurrent theme throughout this dissertation. Different modalities can offer complementary information for a given task. Furthermore, data from existing modalities can be used to replace missing or corrupted information from another source [123]. Therefore, the use of multimodal data generally aids in achieving better solutions, especially

for nontrivial problems. Nevertheless, the way of employing the different modalities for solving a particular task is equally important. From our results, it can be concluded that integral, end-to-end learning approaches are advantageous to combining modalities or training a separate model for each modality and combining the results. As such, it is likely that for many classification and detection problems, including the ones discussed in this thesis, future efforts would be diverted from engineering relevant features to the design of better objective functions as a way of improving the state-of-the-art.

Appendix A

Review of Wildfire Risk Management Using Social Media

With the introduction of social networks and services, there has been an increase of multimodal information sharing on the Internet. The availability of Internet capable mobile devices equipped with various sensors has simplified and liberalized the generation of large amounts of multimedia data. In Chapter 2, we showed how volunteered geographic information can be used for road classification. Over the last few years, however, social media has also played a critical role in some disaster events. This chapter gives a review of current systems and methods that enable the use of social media as a human-centric sensor in large-scale hazard events. We focus particularly on wildfire use-cases and discuss the approaches from other hazard management systems which could be applied in the domain of large area fires. Based on the reviewed systems and methods, we also suggest a general social sensor platform for wildfire detection and management.

A.1 Introduction

Wildfires are one of the leading hazards affecting everyday life around the world. Due to droughts, expansion of the wildland-urban interface, and other factors, the frequency, intensity, and duration of wildfires are increasing worldwide [124]. Recent fires, such as the Rocky fire near San Francisco in California (2015), the Pinery fire (2015) and

the Black Saturday bushfires (2009) in Australia, the Texas bushfires (2011), and the Costa Del Sol fires in Spain (2012), had a big impact on the lives of many of our contemporaries. Because of their speed and destructive forces, they are one of the most serious threats to ecological systems, infrastructure and human lives, i.e., wildfire events that do not involve people and property are becoming rare [125]. To avoid large-scale damage, timely and accurate detection is essential. The sooner the fire is detected, the better the chances are for survival and the lower the environmental and economical impact. However, early detection is not the only crucial aspect of wildfire management. Previously, in Chapters 3 and 4, we have presented methods which can be used to accurately classify different land cover classes from multimodal remote-sensing data. This is important for wildfire risk assessment and prevention. Namely, in case of a wildfire, terrains such as pine and eucalyptus forests can accelerate the fire spread. Furthermore, it is also important to have a clear understanding of the fire development and its location. Where did the fire start? What is the size of the fire? How fast is the fire growing? The answer to each of these questions plays an important part in safety analysis and fire fighting/mitigation, and is essential in assessing the risk of escalation [126]. Up to now, however, information about wildfire circumstances is still rarely available and difficult to measure.

A.1.1 Categorization of wildfire risk management systems

Over the last decade, great efforts have been put into the development of systems for early wildfire detection and wildfire risk management. In addition to the traditional method of human-based wildfire surveillance, shown in Figure A.1 (a), modern automatic wildfire management systems can be further categorized into *terrestrial* and *aerial* systems.

Terrestrial systems are based on information and communication technology (ICT) and camera technology and are becoming more and more important [127–129]. Camera-based wildfire systems offer advanced automatic wildfire observation as a replacement for human observation. As illustrated in Figure A.1 (b), the video-based setup consists of several cameras and/or other sensor devices installed on monitoring spots (watchtowers). A computer system (or a human observer) analyzes the provided video data and generates potential alarms [130]. The main advantages of the camera-based wildfire de-

tection systems are: a wider area that can be covered (because one observer can monitor multiple cameras), zooming options (so that the observer can easily inspect suspected areas), and video storing capabilities for post-fire analysis [131]. Furthermore, heat and fire can be detected in night conditions, if camera systems with infrared sensors are used [132]. Non-visual wireless ICT sensors exist which can be used within dense vegetation. On the negative side, terrestrial systems have to be deployed in networks to be able to cover large areas. Until today, the percentage wildland covered with camera technology and ICT sensors is still limited and time and money (to install these systems) are insufficient.

Aerial systems, on the other hand, such as manned and autonomous aircrafts [133] and satellites (Figure A.2) can cover very large areas. Unlike terrestrially deployed systems, they can be more easily moved and used in affected areas. Aerial systems include a relatively high operational cost and can be affected by visibility conditions.

(a) (b) (c)

Figure A.1: Comparison between traditional human-based wildfire detection (a), camera based wildfire detection (b), and social media based wildfire sensing (c) (based on a scheme proposed by Stipaničev [131]).

In order to be able to fight wildfires in an efficient way, new techniques and systems are needed as a complement to the existing approaches. Social media wildfire sensors, which are illustrated in Figure A.1 (c), are definitely on their way to become one of the wildfire monitoring tools of the future. With the rise of social media, the communication landscape has changed radically. Starting from passive

information dissemination, social media usage is evolving in the direction of active monitoring and proactive public engagement. The fact that social media platforms, such as Facebook, Twitter, Flickr and others, facilitate instantaneous information sharing and are available to anyone-everywhere make them a powerful force. In the context

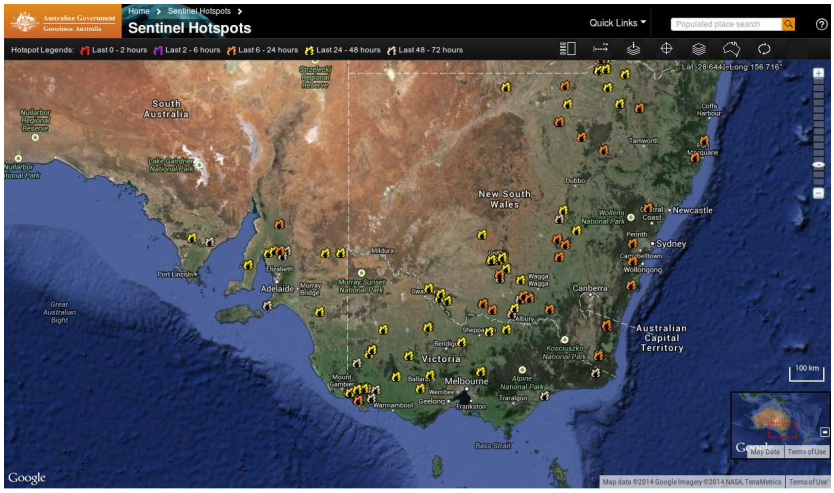


Figure A.2: Surveillance map from the Sentinel Hotspots satellite monitoring system[1]. Recent heat sources (fires, as well as other sources such as industrial furnaces) detected by satellites are displayed on the map.

of crisis communication, for example, social media can be used in order to reach more people, with more relevant and timely information than ever before. On the other hand, social media can also be used to collect (and analyze) user-generated sensor data [134]. This is the idea behind participatory or human-centric sensing. By combining sensor data from large groups of individuals, it is possible to derive new value for end-users in ways that the contributor of the content even did not plan or imagine, and to perform functions that are either difficult to automate or expensive to implement.

A.1.2 Current status of social media in wildfire risk management

Recently, the tendency of participatory data gathering has also started to occur in the domain of disaster analysis. As the number of active

[1] `http://sentinel.ga.gov.au/#/main`

users increases, the data that is produced on social media services can be a valuable additional source of information for prevention, detection, control, and localization of large-scale natural disasters. Furthermore, the ubiquity of mobile communication devices and the fast diffusion of information on social networks render systems for social media information extraction suitable for critical emergency broadcasting. Although the deployment of automatic social media based systems for wildfire management is still at its infancy, relevant agencies for fighting wildfires are beginning to grasp the importance of social media and are progressing towards utilizing the available potential. Typical examples[2,3] include the broadcast of wildfire incidents and emergency warnings through social media networks. However, much more can be gained by using social media in systems for wildfire risk management complementary to traditional solutions. Therefore, in this work we give an overview of the currently available methods that utilize social media in different aspects of disaster event management. In particular, we focus on the approaches in the domain of wildfire detection and management and identify and emphasize components of existing systems for other hazard events which could be put to use in this domain.

The remainder of this appendix is organized as follows. Section A.2 presents a literature overview of the state-of-the-art disaster management methods, applications, and systems using social media information. Subsequently, Section A.3 discusses all steps involved in the social media sensing process. Next, Section A.4 proposes the general architecture of our wildfire social sensor platform, on which we will focus our future research activities. Finally, Section A.5 ends this appendix with conclusions and directions for future work.

A.2 Social media methods, applications, and systems for disaster management

In recent years, social networks and services have been seen not only as a mean for interaction, but also as a human-centric sensor network providing quantifiable, collective intelligence information. A new computing platform, named social computing, emerged to uti-

[2] `https://ruralfire.qld.gov.au/map.html`
[3] `http://www.rfs.nsw.gov.au`

lize the potential of data collected on the Internet. The possibility to employ social media information has proven useful in methods for large-scale disaster management. In this section, we give an overview of such methods, applications, and systems pertinent to wildfire detection and management.

A.2.1 Disaster management methods using social media information

The methods described here address different aspects of the wildfire management problem, such as case studies analyzing the type of shared information and characteristics of social media data during wildfire events [135, 136], extraction of information in disaster related microblog posts [137], diffusion of information in social networks [138], use of social media data in existing physical models for prediction of hazard propagation [139], verification of social media information [95, 140, 141], and visualization of the flow of information [142].

Sutton et al. [135] analyzed the social media communications activity during the October 2007 Southern California Wildfires. Their research showed that social media information had a significant impact during the wildfires crisis. Community feedback from Facebook, Flicker, and mailing lists proved relevant due to the correct information of local geography and the situation in the field. Peer-to-peer (P2P) sharing of news through social media outlets allowed for up-to-date news on the rapidly changing situation when compared to traditional news sources.

Vieweg et al. [136] analyzed Twitter microblog posts during the April 2009 Oklahoma Grassfires for information extraction that could aid common situational awareness. Their analysis showed that 40% of all on-topic tweets in their dataset included some geo-location information (location of people, fires, evacuation sites, highways closed etc). The following tweet from the used dataset illustrates this:

> *Velma area residents: Officials say to take Old Hwy 7 to Speedy G to safely evacuate. Stephens Co Fairgrounds in Duncan for shelter.*

Automatic extraction of relevant geo-location information could be done by employing natural language processing methods for named

entity recognition for microblog posts [13] and geographic entity recognition [143]. Furthermore, 56% of these tweets included updates of the situation in the field. This shows that information systems for support of emergency response teams, as well as situational awareness broadcasting systems, can benefit from the information circulated within the local Twitter community.

Imran et al. [137] propose a method for automatic extraction of information from disaster related microblog posts. The authors used a dataset of tweets sampled during a disaster event (the Joplin 2011 tornado in Joplin, Missouri, USA) to validate their system. Crowdsourcing was used to filter each tweet according to whether the tweet was relevant (informative to some aspect of the disaster) and to label the tweets in four categories: caution and advice, information source, donation, and casualties and damage. Tweets from each category were additionally labeled by sub-categories. For example, tweets in the information source category were labeled by the type of source (web page, photo, video or other) the information in the tweet was coming from. In order to automatically classify a tweet in one of the aforementioned categories, Naive Bayes classifiers were trained on the filtered and annotated dataset. The authors used a number of features for training the classifiers including unigrams, bigrams, and part-of-speech (POS) tags. Finally, different types of information from classified tweets were extracted depending on the category of the tweet. A named entity recognizer and a POS tagger were used to extract information.

Zhu et al. [138] model the diffusion of information in the Twitter network to help optimize emergency communication of messages during disaster events. For a given tweet, the authors try to predict the retweet decision of each of the users in a targeted network. A logistic regression model is used for the retweeting probability of the user given a tweet observation that is projected in feature space. Features are extracted from the observed tweet, and the previous history of the user and target network. The authors identify three categories of features incited by the types of influences that encourage retweeting. Content features are represented by a term frequency vector between a given tweet and previous tweets published by the user, his followers and his friends, and by the presence of URLs, hashtags, and user mentions in the given tweet. Content features reflect the similarities between the content of the incoming tweet and interests related to

the user. Network features such as social influence, mutual followers, friend mentions, and retweets reflect the social relationships between the author of a tweet and the retweeter. Time features are calculated from the average response time of the user and the number of accumulated tweets between responses. Time features include the temporal decay of retweet probability, i.e. the fact that the majority of retweets occur within a time window of the posting of the initial tweet [144].

Social media is used in the work of Aulov and Halem [139] for improving model prediction of the propagation of oil spills on the ocean surface. Since the use of human sensor information in the proposed method is decoupled from the geophysical models, the approach can be applied in the context of wildfire propagation. The authors evaluated their method on data related to the Deepwater Horizon oil spill in the Gulf of Mexico. They used the Flickr API to query images for a period of six months from the time the accident occured. The scope of the queries was restricted by using keywords related to the incident and by retrieving only geotagged images within the area of interest. Metadata attributes related to location (latitude, longitude, and accuracy) and time (date the image was taken) were subsequently extracted from the retrieved set of photos. The obtained locations were superimposed on the model map and used as a ground truth of the model forecast. The authors were able to select suitable parameters (windage and diffusion rate) for the oil propagation model by correlating the model predictions with the ground truth.

Schenk and Sicker [140] propose a ranking algorithm for determining the reputation or the local influence of nodes in dynamic social networks such as Twitter. The authors used Twitter data collected during the Four-mile Canyon Fire, which began in Boulder, Colorado, USA on September 6th 2010. During the incident, they made five one-day snapshots of the social graph of users who posted tweets containing one or more predefined keywords related to the incident. The method then ranks influential users according to the total change in active followers during the event. Other social media can be represented as a social graph, which makes the proposed approach applicable in different environments. Drawbacks of this approach are the requirement for frequent social graph snapshots (updates of the state of the network) and not having a prior distribution of user rankings before the start of the specified event.

Uddin et al. [141] propose a method for source selection in social

sensing applications. The approach aims to increase the credibility of information gathered through social media by eliminating dependant sources, i.e. users who report existing information without prior verification. The authors use the Twitter social graph of a given user to calculate an independence score, which is based on follower-followee relationships with other users. The user is then included in the subset of selected sources if it has an independence score larger than a predefined threshold, and if it improves the sum of independence scores in the subset.

Wang et al. [145] propose a maximum likelihood estimation approach for truth discovery from sensing data, such as mobile sensor data (e.g., text, images, videos, accelerometer data, GPS information), which are shared through social media. The authors deal with data which have the form of multiple binary observations, such as reportings whether an event has occured on multiple locations of interest. They cast the problem into a joint maximum likelihood estimation of source reliability (probability that the observations made by the participant are correct) and observation correctness, which they solve using the expectation-maximization (EM) algorithm. Geotagging simulations performed by the authors show that their approach is accurate in estimating participant reliability and correctness of observations as long as there are multiple independent observations from a given participant (source), and as long as some participants make the same kind of observations. The method outperforms related fact-finder algorithms and common heuristics such as majority voting under the same simulation datasets.

In a recent article, Cao et al. [142] give an overview of visualization methods for social media. The authors also describe a system for real-time visualization of information diffusion in the Twitter network. Namely, visualization of information flows by using retweets. The system visualizes three main entities: tweets, which are represented as topic discs, users grouped by common relations, and diffusion pathways, i.e. curved time lines which represent the path linking a given tweet to the retweet user groups. Visual encodings are used to convey additional information, e.g. color hue encodes sentiment of tweets, activeness of tweets (frequency of retweeting) is encoded through opacity of the corresponding elements, element size is used to depict expected influence (elements for user groups with more followers are bigger), shape discerns tweet user types (such as media

outlets, journalists, or representatives of organizations). Topic discs are rendered by using a sunflower packing algorithm [146] so that inactive tweets are placed in the disk center and active tweets are laid on the edge of the disk in concentric circles. Diffusion pathways are displayed by employing an edge routing algorithm to reduce visual clutter. Pathways are drawn by using an electric field model so that they have similar properties to electrical flux lines (the lines do not cross, and they are as short as possible). The layout of the user groups is similar to the topic discs. Groups are placed on an outer circle either by equally dividing the circle arc or according to the location (longitude) of the retweet users in the group. Tweets and user groups are then represented with a bipartite graph stress model and reordered, by optimizing the model, to reduce line crossings. Finally, line crossings are further reduced by rotating the topic discs according to a spring force model optimization.

A.2.2 Crowdsourcing applications

Crowdsourcing applications harness the power of collective intelligence by delegating collaborative tasks to the community. Because user participation is often on a voluntary basis, these applications are particularly useful for problems which require small contributions from a large number of participants.

Gao et al. [147] describe advantages and shortcomings of crowdsourcing applications for disaster relief coordination. The authors give the example of the Ushahidi[4] crowdsource information platform. The platform can be applied to the wildfire usecase[5], and allows for information collection (via text messages, twitter, email and web forms), visualization of information on an interactive map, and filtering and tracking of data over time. There are several advantages of collecting information from social media through crowdsourcing applications. The data is available in real time and may contain geo-location information, which enables disaster relief teams to pinpoint the areas that require immediate attention. Coupled with proactive analysis of micro post information could help with the discovery of most-critical categories to better direct resources and relief efforts. On the downside, such social media applications for disaster relief do not provide

[4] `www.ushahidi.com`

[5] `https://openforesteitaliane.crowdmap.com/`

explicit mechanisms that would guarantee coordination between relief organizations and efficient allocation of resources. Furthermore, the flow of information should be organized in a way which would ensure the safety of response teams. Finally, since the time of response is critical in disaster events, there needs to be a way of verifying the accuracy of the gathered information.

Vivacqua and Borges [148] consider the use of collective intelligence and crowdsourcing in the domain of emergency response. The authors have identified that many crisis response organizations do not have the necessary human resources for extensive research of the information necessary in time of disasters. A problem that fire departments face is the lack of information about the area surrounding the location of the emergency. The authors propose solutions for the aforementioned problem which are based on collective intelligence. A database of functional fire hydrants can be created by crowdsourcing the work to the community. In this way the users can create and update the database by marking fire hydrants from their neighborhood on online maps.

Figure A.3: FireMash, an application for reporting wildfires in New South Wales, Australia via Twitter messages [148].

Another example of a crowdsourcing initiative is *FireMash*—a system for reporting wildfires in New South Wales, Australia (see Figure A.3). The system combines live feeds (RSS and Twitter) from the New South Wales Rural Fire Service and maps the fires on a Google

Map. Citizens report wildfires by tweeting the fire location. The system also provides local notices of fires that are close to the location of a given user.

A.2.3 Social media disaster management systems

In comparison with the methods and applications discussed previously in Sections A.2.1 and A.2.2, social media disaster management systems combine multiple techniques in order to provide relevant information in hazard events, and offer a higher level of coordination or automation. We describe several examples of such systems.

A system which uses social media for increasing emergency situation awareness is given by Yin et al. [149]. The system uses the Twitter APIs to capture data streams from a region of interest. In order to detect incident events, the captured tweets are processed by calculating the probability that a number of tweets contain a keyword of interest (such as "fire") within a time window according to a binomial distribution. Then, if in a given time window the frequency of tweets containing the word "fire" is higher than the previously calculated probability, a fire incident is detected. The system further classifies disaster impact tweets (ones that contain information about infrastructure status) by using a support vector machine classifier over a space of textual features extracted from the tweets (unigrams, bigrams, word length, number of hashtags contained in the tweet, number of user mentions, whether the tweet has been retweeted, and whether a tweet has been replied to by other users). Tweets are then clustered by topic. Each tweet is represented by a term frequency-inverse document frequency vector in an online clustering algorithm where the tweets are grouped according to their Jaccard similarity and time of publishing. Finally, geographic locations of tweets are tagged on an interactive map for easy visualization.

In the work of Patrikakis et al. [150], a social networking and P2P based multimedia streaming framework for assistance in rescue and relief operations is discussed. In the case of a forest fire, an emergency operation support system could facilitate the flow of information between the involved parties. The aim of the proposed approach is to create social network groups of volunteers consisting of experts and practitioners from various fields. The profile of each of the members of a group would include their location, skills (medical knowledge, fire fighting knowledge, languages spoken) and resources (equipment

and vehicles at disposal). In case of a fire, resources and people can be grouped and dispatched according to their locality and availability. The P2P platform would allow for people who live in the fire affected area to use their mobile devices to capture and distribute videos of potentially hazardous situations with corresponding annotations. The annotations would provide additional metadata, such as the location of the scene in the video, the time, and meteorological conditions in the field. This community provided information would then be effectively used in fire fighting and evacuation operations.

Nabil et al. [151,152] give an overview of a system for Social Media Alert and Response to Threats to Citizens (SMART-C). The system consists of three components. The data collection component provides support for resilient capture and retrieval from heterogeneous information sources (voice, text, image, and video). Data enrichment comprises a subsequent part of the system used to extract semantic information from data. For event detection in videos, the authors used a SVM approach and low-level features [153] aggregated over the video sequence into a bag-of-words representation. The last system component enables customized alerting. This component consists of server side modules for generating context based alerts for subscribers, and a client side mobile application. The context-based alerts may appertain to location, connectivity, language spoken, and other specific characteristics of the targeted group. For example, people living in areas close to storing facilities for hazardous chemicals would receive a specialized alert in case of a fire.

A similar Global Disaster Alert and Coordination System (GDACS) is described in the work of Stollberg and de Groeve [154]. The system aims to provide global multihazard disaster alerting by using social media for dissemination, monitoring, and exchange of information. In this approach, the Twitter stream is continuously monitored, and messages containing keywords (such as "earthquake") are stored for near real-time detection of global disaster events. Events are detected by locating peaks of tweets mentioning a relevant keyword. Alert information is then propagated to the public through a mobile application. The application allows for early responders to send reports of a disaster event. The system automatically geolocates photos from reports and indicates the locations on a map for further assesment.

Sakaki et al. [155] propose a near real-time system for detection and reporting of disaster events. They focus on the detection of earth-

quakes and estimation of typhoon trajectories from Twitter microblog posts, however, the proposed system can also be used for detection of other large-scale hazard events, such as wildfires. In this approach, event detection is performed first by classification of positive instances of real-time event microposts (such as the occurrence of an earthquake) using a trained SVM classifier. To train the classifier, the presence of keywords is used as features together with the number of words in the micropost, the position of the keyword within the message, and the words before and after the keyword. Then, taking into account the posting time of the positive event microposts, a temporal model following an exponential distribution is constructed. Given a number of positive event postings, the model allows to estimate the probability of an alarm occurrence with a low false-positive ratio. Each micropost is further regarded as a sensor reading associated with a location (only microposts containing GPS data or a registered user location are used). By using a particle filter [156], a spatial model is created, which allows to estimate the location of the event from the sensor readings. In the end, a warning email is sent to registered users when an earthquake is detected. The proposed system was compared to the early warning service (broadcasted on television) of the Japan Meteorological Agency (JMA), and it was able to detect 96% of earthquakes with a JMA intensity scale 3 or higher while delivering alarms far faster than the JMA announcements (emails were sent by the system mostly within 1 min. compared to 6 min. for the JMA announcement) [155]. In the context of wildfire risk management, the usefulness and applicability in wildfire detection of this kind of social media based system is apparent considering the capability for providing time-sensitive information, and also its complementarity and cost efficiency compared to traditional ICT and sensor based systems.

Given the overview of the state of the art works in large-scale disaster management, in the next section, we discuss in detail the process of human-centric sensing using social media.

A.3 Social media data management—the sensing process

Within traditional wildfire detection systems the role of humans is mostly limited to being end-consumers of information. In human-centric sensing systems (such as the social media based disaster man-

agement systems presented in the previous section) a larger involvement of humans is needed along other points in the data-to-decision path. This path generally consists of sensing, i.e., acquisition of sensor measurements, and information processing [134]. The processing mainly focuses on extracting relevant information and metadata from the sensor measurements, and fusing and analyzing such information from multiple sources to derive knowledge that forms the basis of fire fighting decisions and actions. Goldman et al. [157] split up this participatory sensing process into 8 steps (coordination, capture, transfer, storage, access, analysis, feedback, and visualization), where each of the steps are facilitated by a corresponding technology. Hereafter, we discuss each of these steps within the context of wildfire detection.

Coordination involves recruiting and communicating with participants to explain the sensing effort and provide necessary guidance. In the context of fire detection, an example of this can be given by *FireMash*—a system to report bushfires in Australia [148]. The system allows citizens to report a fire by using a specific hashtag (#nswfires) as a form of coordination. Important to remark is that attention must be paid in defining the crowdsource. A participatory sensing application will only be as good as the participants supporting it. Recruitment of participants in sensing campaigns will be a determinant factor for the success of their outcome [158]. A social media sensor developer should identify the kind of information required and the best set of participants to provide that information. Tools and techniques are needed that can support discovery, recruitment, and maintenance of the set of participants [159].

Capture is the acquisition of data by the users of the system. Current social media services and networks facilitate sharing information in various formats such as text messages, images, and videos and by using mobile communication devices. When collecting this data, it is of high importance to address the range of potential underlying possibilities for sensor failures and user errors, both accidental and malicious in intent [160]. Some data will be more useful than others, and some sort of pre-evaluation or filtering may be necessary. Direct evaluation of data by users is, as such, not always possible and the capturing process should be supported by some tools and algorithms at a lower level [158]. The SocialSensor[6] project, for example, is developing real-time focused crawling and data mining techniques for

[6] `http://www.socialsensor.eu/`

extracting useful information in terms of topics, events, spatial and temporal trends, combining multimodal data coming from various online social networks. While the use cases of the project are related to the news and infotainment domains, most of the techniques are generic enough to be applied to other areas, such as the fire detection and response use cases. Related to this, it can also be useful to study the outcome of the WeKnowIt[7] project, with objectives directly related to emergency response through user-generated content analysis. Finally, Twitcident [161] is a framework that offers automatic filtering and tracking of semantically enriched Twitter streams by monitoring emergency broadcasting services.

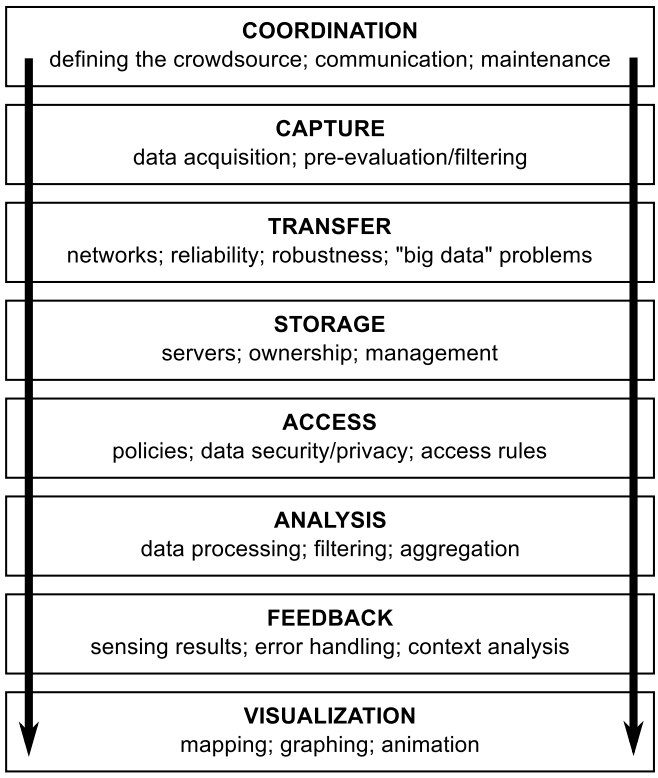


Figure A.4: Diagram of the sensing process for social media data management.

Transfer takes place using different communication networks, including both wired and wireless communication. Wireless mobile

[7] `http://www.weknowit.eu/`

communications can be set up with minimal infrastructure (e.g., ad-hoc networks) and in a way which allows sufficient redundancy for reliable accessibility even in the event of disasters. When it comes to systems for wildfire scenarios, the ease of installation, reliability and robustness is of utmost importance. It is important to provide reliable and robust communication frameworks, focusing on the technical aspects of sensor data communication in a forest fire environment.

It is also important to cope with the fact that social sensor systems collect large amounts of data, which must be continuously stored and processed. Furthermore, the number of users in a social network can be very large, leading to scalability challenges for the storage and processing of the underlying streams [162]. Both "big data" problems definitely need to be adequately addressed in the upcoming years.

Storage occurs on servers distributed across the Internet. These can be privately owned and commercially managed servers which allow secure, private access storage services.

Access is managed according to policies written by project organizers and participants. It is difficult to overstate the importance and intricacy of data security and privacy. Currently, many people entrust their private e-mail and other data to website providers because standard access controls make them comfortable that their accounts are safe and private. It is also common for people to choose to share their information with other trusted members of a network according to a specific and user-controlled set of rules.

Analysis includes a wide variety of data-processing methods, from aggregation and filtering of contributed data to automated report summarization, extraction of geo-locations and higher-level analysis of data for prediction of social behavior and modeling of fire propagation. Analysis also includes verification of the contributed data and the calculation of group statistics.

Feedback can be provided to affected people and groups in a context-dependent way. For example, evacuation information can be broadcasted during a wildfire disaster to endangered areas based on predicted weather conditions and the on-site situation (strong winds, inaccessibility of road infrastructure etc). In order to improve the sensing process, information regarding the sensing results and data errors should also be sent back to the users.

Visualization goes hand-in-hand with analysis and is the step in which data are displayed in a legible format to relevant parties (re-

sponse teams, organizational units, people affected by the disaster). The effectiveness of any project depends on how well its results are understood by the target audience. Excellent methods for mapping, graphing, and animation make this a rich area to develop in the context of wildfire detection and management.

A.4 Wildfire social sensor platform

In Section A.2, several methods were described that addressed different aspects of the information flow in disaster management. Based on practices of the existing techniques, and taking into account the different stages of human-centric sensing, as given in Section A.3, we identify the necessary components of a social sensor platform for wildfire management. The proposed design is intended as a general architecture which would enable effective use of social computing while allowing to target different specific aspects of wildfire management, such as fire detection, emergency response, resource allocation, and fire localization.

Collection is the initial stage of gathering information from social media. It incorporates the coordination and capture stages of participatory sensing. There are, in general, two ways to collect social media information from the Internet. In a passive listening approach, there is no prior coordination between the users about the format of the data, and there is no central point, such as a dedicated social network profile, where users can send information. On the other hand, an active listening approach offers at least a form of coordination (e.g., a reserved hashtag for Twitter messages intended for the system), or a central point for getting data through to the system. The advantage of the latter approach is that it facilitates to an extent the step of information verification, while the benefit from the former approach is a larger scope of user data. Since active and passive data collection are complementary, they can be combined to increase the effectiveness of the system in this stage.

Aggregation and filtering deals with grouping of different types of social media data and filtering redundant information from the gathered dataset. Although these tasks can themselves be objects of data analysis, due to the typically large quantities of data in social media, and the real-time requirements of wildfire management systems, they are usually implemented as a dedicated system module. The most

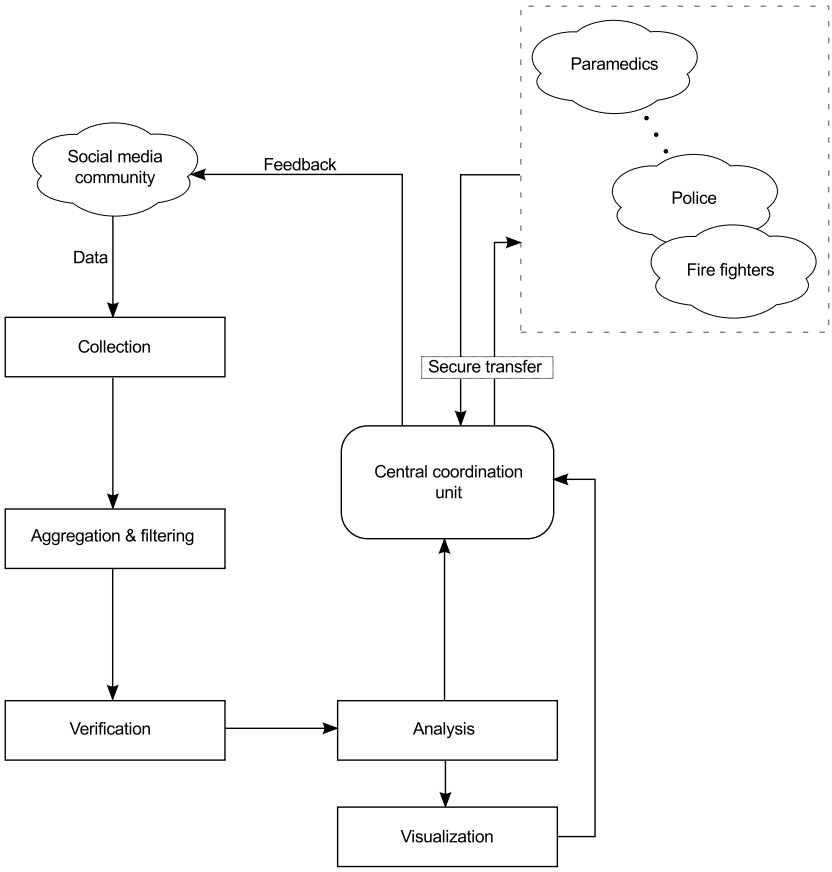


Figure A.5: Diagram of a wildfire management social sensor platform.

used types of information in the context of the problem are text, geographical data, image, and video data. There are existing methods for aggregation and filtering of the specified data categories [163–165]. Furthermore, this part of the platform can be augmented by using the different modalities offered by social media services and networks. Namely, many social media, such as Twitter, Facebook, YouTube, and Flickr, offer application programming interfaces with possibilities for posting specialized queries to retrieve specific groups of information.

Verification is necessary for estimating the accuracy of the collected data. It is particularly important because of the sensitivity of the information that pass through the system to different organizations involved in a wildfire crisis management. Different statistical techniques and inference methods, such as the maximum likelihood estimation approach [145], could be used for information verification. However, unbiased sampling of sources [141] is required to mitigate the effect of false information diffusion in social networks.

Analysis is the part of the system that is responsible for automatic processing of collected data and inferring relevant, high level information. For example, a subsystem could analyze geotag locations of social media reportings of a wildfire to infer the envelope of the area caught by fire, and maintain a model of fire propagation based on information of meteorological conditions in the field.

Visualization is tightly coupled with analysis, and serves to enable a more intuitive representation of the analysis results to a human operator or system user. A dominant trend of visualization in large-scale disaster management is the use of geographic map mashups. Other visualization schemes beneficial in wildfire detection could include automatic registration systems for aerial and satellite images.

Central coordination provides for secure and timely exchange of information between the individuals, groups and organizations involved in the incident. Due to the ubiquity of mobile devices and their networking capabilities, as well as the fast propagation of information in social media, it would be very useful for a disaster management system to provide a feedback loop to social media communities. Such an information channel should be provided by an official crisis management body, as is the case of a central coordination unit, which would allow to fully exploit the versatility offered by human sensor networks.

A.5 Conclusions and original contributions

Wildfires large-area natural disasters for which early detection is necessary in order to achieve effective localization. Social media could be used here to provide a kind of human-centric sensor network for early detection of such fires. The collective intelligence information gathered from these kinds of sources could also be used to coordinate response teams, efficiently allocate resources and could serve to increase situational awareness. Studies discussed previously show that there are existing methods which can be used in a system for detection and management of large area fires. Moreover, our review on the use of social media methods has contributed toward the potential of using collective intelligence information in different hazard approaches [166–170]. Additionally, in this appendix, we have proposed a general architecture for a wildfire social sensor platform which can serve as a guideline for effective use of social media information in systems for wildfire management. Further research could be done in the direction of system development for improving coordination between different organizational units involved in a disaster event. Another aspect that is missing or that is not fully exploited in existing systems, is proactive exchange of information with social media users and contributors. Therefore, in order to exploit the full benefits offered by social media channels, a reliable mechanism providing a feedback loop between official response organizations and the social media community would have to be developed.

The work presented in this appendix is published in *Fire Safety Journal*:

- Viktor Slavkovikj, Steven Verstockt, Sofie Van Hoecke, and Rik Van de Walle. Review of wildfire detection using social media. *Fire Safety Journal, vol. 68, no. 0, pp. 109–118, 2014.*

Bibliography

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] Q. Summerfield, "Lipreading and audio-visual speech perception," *Philosophical Transactions: Biological Sciences*, vol. 335, no. 1273, pp. 71–78, 1992.

[3] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, ser. Wiley series in remote sensing. Hoboken, N.J., Chichester: Wiley, 2003.

[4] H.-C. Lee, *Introduction to Color Imaging Science*. New York, NY, USA: Cambridge University Press, 2005.

[5] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, Jan 1968.

[6] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for twitter hashtag recommendation," in *Proceedings of the 22Nd International Conference on World Wide Web Companion*, ser. WWW '13 Companion, 2013, pp. 593–596.

[7] O. Janssens, R. Schulz, V. Slavkovikj, K. Stockman, M. Loccufier, R. V. de Walle, and S. V. Hoecke, "Thermal image based fault diagnosis for rotating machinery," *Infrared Physics & Technology*, vol. 73, pp. 78 – 87, 2015.

[8] O. Janssens, L. Verlendens, R. Schulz, V. Ongenae, K. Stockman, M. Loccufier, R. Van de Walle, and S. Van Hoecke, "Infrared and vibration based bearing fault detection using neural networks," in *13th International Workshop on Advanced Infrared Technology and Applications, Proceedings*, 2015, pp. 220–223.

[9] S. Verstockt, T. Beji, P. De Potter, S. Van Hoecke, B. Sette, B. Merci, and R. Van de Walle, "Video driven fire spread forecasting (f) using multi-modal lwir and visual flame and smoke data," *PATTERN RECOGNITION LETTERS*, vol. 34, no. 1, pp. 62–69, 2013.

[10] S. Verstockt, S. Van Hoecke, P. De Potter, P. Lambert, C. Hollemeersch, B. Sette, B. Merci, and R. Van de Walle, "Multi-modal time-of-flight based fire detection," *MULTIMEDIA TOOLS AND APPLICATIONS*, vol. 69, no. 2, pp. 313–338, 2014.

[11] S. Verstockt, R. Van de Walle, L. A. Gonzalez Avila, B. Merci, and J. De Blonde, "Combining volume sensors with multi-modal video analysis for fire detection and forecasting," in *International Conference on Automatic Fire Detection, Proceedings*, vol. 2, 2014, pp. 203–210.

[12] F. Godin, B. Vandersmissen, A. Jalalvand, W. De Neve, and R. Van de Walle, "Alleviating manual feature engineering for part-of-speech tagging of Twitter microposts using distributed word representations," in *Workshop on Modern Machine Learning and Natural Language Processing, NIPS 2014 Proceedings*, 2014, p. 5.

[13] F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Multimedia lab @ ACL W-NUT NER sharedtask: named entity recognition for Twitter microposts using distributed word representations," in *ACL 2015 Workshop on Noisy User-generated Text, Proceedings*. Association for Computational Linguistics, 2015, pp. 146–153.

[14] B. Vandersmissen, F. Godin, A. Tomar, W. De Neve, and R. Van de Walle, "The rise of mobile and social short-form video: an in-depth measurement study of vine," in *CEUR workshop*

proceedings, S. Papadopoulos, P. C. Cesar, D. A. S. Shamma, A. Kelliher, and R. Jain, Eds., vol. 1198, 2014, pp. 1–10.

[15] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

[16] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava, "Biketastic: sensing and mapping for better biking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1817–1820.

[17] C. Weiss, H. Fröhlich, and A. Zell, "Vibration-based terrain classification using support vector machines." in *IROS*. IEEE, 2006, pp. 4429–4434.

[18] C. C. Ward and K. Iagnemma, "Speed-independent vibration-based terrain classification for passenger vehicles," *Vehicle System Dynamics*, vol. 47, no. 9, pp. 1095–1113, 2009.

[19] D. Popescu, R. Dobrescu, and D. Merezeanu, "Road analysis based on texture similarity evaluation," in *Proceedings of the 7th WSEAS International Conference on Signal Processing*, ser. SIP'08. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008, pp. 47–51.

[20] I. Tang and T. Breckon, "Automatic road environment classification," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 476–484, 2011.

[21] Y. Khan, P. Komma, and A. Zell, "High resolution visual terrain classification for outdoor robots," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1014–1021.

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[23] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 13:1–13:27, Mar. 2010.

[24] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, ser. IAAI'05. AAAI Press, 2005, pp. 1541–1546.

[25] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds., vol. 3001. Springer Berlin Heidelberg, 2004, pp. 1–17.

[26] A. Pinheiro, "Image descriptors based on the edge orientation," in *Semantic Media Adaptation and Personalization, 2009. SMAP '09. 4th International Workshop on*, 2009, pp. 73–78.

[27] S. F. Ershad, "Texture classification approach based on combination of edge & co-occurrence and local binary pattern," *CoRR*, vol. abs/1203.4855, 2012.

[28] M. Pietikäinen, G. Zhao, A. Hadid, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, ser. Computational Imaging and Vision. Springer, 2011, no. 40.

[29] J. Gall, N. Razavi, and L. Van Gool, "An introduction to random forests for multi-class object detection," in *Proceedings of the 15th International Conference on Theoretical Foundations of Computer Vision: Outdoor and Large-scale Real-world Scene Analysis*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 243–263.

[30] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1297–1304.

[31] J. Shotton, T.-K. Kim, and B. Stenger, "Boosting & randomized forests for visual recognition (tutorial)," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[32] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning

of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. NY, USA: ACM, 2009, pp. 609–616.

[33] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *Journal of Machine Learning Research - Proceedings Track*, vol. 15, pp. 215–223, 2011.

[34] CIE, "Colorimetry," CIE Publication No. CIE 15.2, Central Bureau of the CIE, Vienna, 1986.

[35] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[36] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 1177–1178.

[37] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[38] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th International Conference on Machine Learning*, ser. ICML '11. NY, USA: ACM, June 2011, pp. 921–928.

[39] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[40] G. Strazdins, A. Mednis, R. Zviedris, G. Kanonirs, and L. Selavo, "Virtual ground truth in vehicular sensing experiments: how to mark it accurately," *Proceedings of 5th International Conference on Sensor Technologies and Applications (SENSORCOMM 2011)*, pp. 295–300, 2011.

[41] Y. Khan, A. Masselli, and A. Zell, "Visual terrain classification by flying robots," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 498–503.

[42] X. Sun, Q. Qu, N. Nasrabadi, and T. Tran, "Structured priors for sparse-representation-based hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 7, pp. 1235–1239, July 2014.

[43] A. Charles, B. Olshausen, and C. Rozell, "Learning sparse codes for hyperspectral imagery," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 5, pp. 963–978, Sept 2011.

[44] A. Castrodad, Z. Xing, J. B. Greer, E. Bosch, L. Carin, and G. Sapiro, "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4263–4281, 2011.

[45] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[46] C.-I. Chang, Q. Du, T.-L. Sun, and M. L. G. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2631–2641, 1999.

[47] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1586–1600, 2006.

[48] S. Kaewpijit, J. Le-Moigne, and T. El-Ghazawi, "Automatic reduction of hyperspectral imagery using wavelet spectral analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 4, pp. 863–871, April 2003.

[49] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, p. 2319, 2000.

[50] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

[51] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NIPS*. MIT Press, 2001, pp. 585–591.

[52] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[53] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proceedings of the International Conference on Computer Vision (ICCV)*, vol. 2, Oct 2005, pp. 1208–1213.

[54] X. He and P. Niyogi, "Locality preserving projections," in *NIPS*. MIT Press, 2004, pp. 153–160.

[55] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, no. 7-9, pp. 1547–1553, 2007.

[56] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331–341, 2010.

[57] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Oct 2007, pp. 1–7.

[58] S. Chen and D. Zhang, "Semisupervised dimensionality reduction with pairwise constraints for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 2, pp. 369–373, 2011.

[59] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local Fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, no. 1-2, pp. 35–61, 2010.

[60] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 184–198, 2013.

[61] Z. Shao and L. Zhang, "Sparse dimensionality reduction of hyperspectral image based on semi-supervised local Fisher discriminant analysis," *International Journal of Applied Earth Observation and Geoinformation*, vol. 31, pp. 122–129, sep 2014.

[62] H.-Y. Huang and B.-C. Kuo, "Double nearest proportion feature extraction for hyperspectral-image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4034–4046, 2010.

[63] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1027–1061, may 2007.

[64] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[65] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009, pp. 1794–1801.

[66] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010, pp. 3360–3367.

[67] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*. MIT Press, 2006, pp. 801–808.

[68] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 689–696.

[69] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[70] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, Jul. 2011.

[71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[72] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[73] J. Li, X. Huang, P. Gamba, J. Bioucas-Dias, L. Zhang, J. Atli Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1592–1606, March 2015.

[74] J. Benediktsson, J. Palmason, and J. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 3, pp. 480–491, March 2005.

[75] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[76] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986.

[77] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[78] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[79] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *Journal of Physiology (London)*, vol. 195, pp. 215–243, 1968.

[80] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, 2009.

[81] J. Li and Y. Qian, "Dimension reduction of hyperspectral images with sparse linear discriminant analysis," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2011, pp. 2927–2930.

[82] B.-C. Kuo and D. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[83] B.-C. Kuo, H.-H. Ho, C.-H. Li, C.-C. Hung, and J.-S. Taur, "A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 1, pp. 317–326, Jan 2014.

[84] Y. Chen, N. Nasrabadi, and T. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, Oct 2011.

[85] S. Yang, H. Jin, L. Yang, W. Xu, and L. Jiao, "Compressive sensing-inspired dual-sparse SLFNN for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 220–224, Jan 2014.

[86] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 6, pp. 2094–2107, June 2014.

[87] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Image Processing (ICIP), 2014 IEEE International Conference on*, Oct 2014, pp. 5132–5136.

[88] Y. Chen, X. Zhao, and X. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. PP, no. 99, pp. 1–12, 2015.

[89] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification,"

IEEE Trans. Geoscience and Remote Sensing, vol. 54, no. 3, pp. 1349–1362, 2016.

[90] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, Jul. 2015.

[91] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2015, pp. 4959–4962.

[92] Y. Liu, G. Cao, Q. Sun, and M. Siegel, "Hyperspectral classification via deep networks and superpixel segmentation," *International Journal of Remote Sensing*, vol. 36, no. 13, pp. 3459–3482, 2015.

[93] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.

[94] Y. Lecun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2006, pp. 739–746.

[95] T. Wang, D. Wu, A. Coates, and A. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 3304–3308.

[96] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.

[97] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6964–6968.

[98] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[99] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 2010, pp. 807–814.

[100] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[101] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, vol. 28, no. 3, May 2013, pp. 1139–1147.

[102] E. J. Terrell, W. M. Needelman, and J. P. Kyle, "Wind turbine tribology," in *Green Tribology*, ser. Green Energy and Technology, M. Nosonovsky and B. Bhushan, Eds. Springer Berlin Heidelberg, 2012, pp. 483–530.

[103] J. Lacey, "An overview of bearing vibration analysis," *Maintenance & asset management*, vol. 23, no. 6, pp. 32–42, 2008.

[104] P. Boškoski, J. Petrovčič, B. Musizza, and Đani Juričić, "Detection of lubrication starved bearings in electrical motors by means of vibration analysis," *Tribology International*, vol. 43, no. 9, pp. 1683 – 1692, 2010.

[105] M. Monte, F. Verbelen, and B. Vervisch, "The use of orbitals and full spectra to identify misalignment," in *IMAC XXXII, Proceedings*. Springer International Publishing, 2014, pp. 215–222.

[106] B. P. Graney and K. Starry, "Rolling element bearing analysis," *Materials Evaluation*, vol. 70, no. 1, pp. 78 – 85, 2012.

[107] R. Heng and M. Nor, "Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition," *Applied Acoustics*, vol. 53, no. 1–3, pp. 211 – 226, 1998.

[108] H. Ohta, Y. Nakajima, S. Kato, and H. Tajimi, "Vibration and acoustic emission measurements evaluating the separation of the balls and raceways with lubricating film in a linear bearing under grease lubrication," *Journal of Tribology*, vol. 135, no. 4, 2013.

[109] A. Purarjomandlangrudi, A. H. Ghapanchi, and M. Esmalifalak, "A data mining approach for fault diagnosis: An application of anomaly detection algorithm," *Measurement*, vol. 55, pp. 343 – 352, 2014.

[110] O. Geramifard, J. Xu, C. Pang, J. Zhou, and X. Li, "Data-driven approaches in health condition monitoring – a comparative study," in *8th IEEE International Conference on Control and Automation (ICCA)*, June 2010, pp. 1618 – 1622.

[111] Q. Miao, D. Wang, and M. Pecht, "A probabilistic description scheme for rotating machinery health evaluation," *Journal of Mechanical Science and Technology*, vol. 24, no. 12, pp. 2421 – 2430, 2010.

[112] A. Verma, Z. Zhang, and A. Kusiak, "Modeling and prediction of gearbox faults with data-mining algorithms," *Journal of Solar Energy Engineering*, vol. 135, no. 3, pp. 1 – 11, 2013.

[113] D. Kateris, D. Moshou, X.-E. Pantazi, I. Gravalos, N. Sawalhi, and S. Loutridis, "A machine learning approach for the condition monitoring of rotating machinery," *Journal of Mechanical Science and Technology*, vol. 28, no. 1, pp. 61 – 71, 2014.

[114] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Applied Acoustics*, vol. 89, pp. 16 – 27, 2015.

[115] P. Kankar, S. C. Sharma, and S. Harsha, "Fault diagnosis of ball bearings using machine learning methods," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1876 – 1886, 2011.

[116] J. Fitch, "The hidden dangers of lubricant starvation," http://goo.gl/XaBg4, 2012, online; accessed 18 November 2015.

[117] C. Wang, M. Gan, and C. Zhu, "Fault feature extraction of rolling element bearings based on wavelet packet transform and sparse representation theory," *Journal of Intelligent Manufacturing*, pp. 1 – 15, 2015.

[118] S. Deng, B. Jing, S. Sheng, Y. Huang, and H. Zhou, "Impulse feature extraction method for machinery fault detection using fusion sparse coding and online dictionary learning," *Chinese Journal of Aeronautics*, vol. 28, no. 2, pp. 488 – 498, 2015.

[119] N. Verma, V. Gupta, M. Sharma, and R. Sevakula, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," in *IEEE Conference on Prognostics and Health Management (PHM)*, 2013, pp. 1 – 7.

[120] Schaeffler Technologies GmbH & Co., "Split plummer block housings SNV," http://goo.gl/ctLWuu, March 2010, online; accessed 20 November 2015.

[121] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[122] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 818–833.

[123] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 3128–3137.

[124] A. E. Cetin, D. Akers, I. Aydin, N. Dogan, O. Günay, and B. U. Toreyin, "Using Surveillance Systems for Wildfire Detection," http://goo.gl/0nArzz, 2013, online; accessed 02 May 2014.

[125] D. Stipaničev, M. Štula, D. Krstinić, L. Šerić, T. Jakovčević, and M. Bugarić, "Advanced automatic wildfire surveillance and monitoring network," in *6th International Conference on Forest Fire Research*, no. A20, Nov. 2010.

[126] S. Verstockt, "Multi-modal video analysis for early fire detection," Ph.D. dissertation, Ghent University, 2011.

[127] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, "Real-time wildfire detection using correlation descriptors," in *19th European Signal Processing Conference*, Sep 2011, pp. 894–898.

[128] J. R. Martinez de Dios, B. C. Arrue, A. Ollero, L. Merino, and F. Gómez-Rodríguez, "Computer vision techniques for forest fire perception," *Image Vision Comput.*, vol. 26, no. 4, pp. 550–562, 2008.

[129] O. Günay, K. Taşdemir, B. U. Töreyin, and A. E. Çetin, "Video based wildfire detection at night," *Fire Safety Journal*, vol. 44, no. 6, pp. 860–868, 2009.

[130] T. Jakovčević, L. Šerić, D. Stipaničev, and D. Krstinić, "Wildfire Smoke-Detection Algorithms Evaluation," in *Proceedings of International Conference on Forest Fire Research*, nov 2010.

[131] D. Stipaničev, "Intelligent forest fire monitoring system-from idea to realization," pp. 58–73, 2012.

[132] A. E. Cetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Gunay, Y. H. Habiboglu, B. U. Toreyin, and S. Verstockt, "Video fire detection: review," *DIGITAL SIGNAL PROCESSING*, vol. 23, no. 6, pp. 1827–1843, 2013.

[133] Airbeam, "AIRborne information for Emergency situation Awareness and Monitoring," http://airbeam.eu/project/, 2012, online; accessed 5 Mar. 2016.

[134] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing." *Philos Transact A Math Phys Eng Sci*, vol. 370, no. 1958, pp. 176–97, 2012.

[135] J. N. Sutton, L. Palen, I. Shklovski, and F. i. I. conference, "Backchannels on the front lines : emergency uses of social media in the 2007 Southern California wildfires." in *Proceedings of the 5th International ISCRAM Conference*, May 2008.

[136] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1079–1088.

[137] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, "Extracting information nuggets from disaster-related messages in social media," in *Proceedings of the 10th International ISCRAM Conference*, 2013.

[138] J. Zhu, F. Xiong, D. Piao, Y. Liu, and Y. Zhang, "Statistically modeling the effectiveness of disaster information in social media," in *Proceedings of the 2011 IEEE Global Humanitarian Technology Conference*, ser. GHTC '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 431–436.

[139] O. Aulov and M. Halem, "Human sensor networks for improved modeling of natural disasters," *Proceedings of the IEEE*, vol. 100, no. 10, pp. 2812–2823, 2012.

[140] C. B. Schenk and D. C. Sicker, "Finding event-specific influencers in dynamic social networks," in *SocialCom/PASSAT*. IEEE, 2011, pp. 501–504.

[141] M. Uddin, M. Amin, H. Le, T. Abdelzaher, B. Szymanski, and T. Nguyen, "On diversifying source selection in social sensing," in *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, june 2012, pp. 1 –8.

[142] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu, "Whisper: Tracing the spatiotemporal process of information diffusion in real time," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2649–2658, 2012.

[143] J. L. Leidner and M. D. Lieberman, "Detecting geographical references in the form of place names and associated spatial natural language," *SIGSPATIAL Special*, vol. 3, no. 2, pp. 5–11, Jul. 2011.

[144] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *WWW '10: Proceedings of the 19th international conference on World Wide Web*. New York, NY, USA: ACM, 2010, pp. 591–600.

[145] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: a maximum likelihood estimation approach," in *Proceedings of the 11th international conference*

on Information Processing in Sensor Networks, ser. IPSN '12. New York, NY, USA: ACM, 2012, pp. 233–244.

[146] H. Vogel, "A better way to construct the sunflower head," *Mathematical Biosciences*, vol. 44, no. 3-4, pp. 179–189, 1979.

[147] H. Gao, G. Barbier, and R. Goolsby, "Harnessing the crowdsourcing power of social media for disaster relief," *Intelligent Systems, IEEE*, vol. 26, no. 3, pp. 10–14, 2011.

[148] A. Vivacqua and M. Borges, "Collective intelligence for the design of emergency response," in *Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference on*, 2010, pp. 623–628.

[149] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power, "Using social media to enhance emergency situation awareness," *Intelligent Systems, IEEE*, vol. 27, no. 6, pp. 52–59, 2012.

[150] C. Patrikakis, A. Voulodimos, E. Sardis, N. Papaoulakis, D. Christofi, and G. Dimosthenous, "Emergency operations support through social networking and p2p multimedia services," in *Telecommunications (ICT), 2011 18th International Conference on*, 2011, pp. 124–129.

[151] N. Adam, J. Eledath, S. Mehrotra, and N. Venkatasubramanian, "Social media alert and response to threats to citizens (smart-c)," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, 2012, pp. 181–189.

[152] N. Adam, B. Shafiq, and R. Staffin, "Spatial computing and social media in the context of disaster management," *Intelligent Systems, IEEE*, vol. 27, no. 6, pp. 90–96, 2012.

[153] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 3681–3688.

[154] B. Stollberg and T. de Groeve, "The use of social media within the global disaster alert and coordination system (GDACS)," in

Proceedings of the 21st international conference companion on World Wide Web, ser. WWW '12 Companion. New York, NY, USA: ACM, 2012, pp. 703–706.

[155] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 851–860.

[156] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *Pervasive Computing, IEEE*, vol. 2, no. 3, pp. 24–33, July 2003.

[157] J. Goldman, K. Shilton, J. Burke, D. Estrin, M. Hansen, N. Ramanathan, S. Reddy, V. Samanta, M. Srivastava, and R. West, "Participatory sensing: A citizen-powered approach to illuminating the patterns that shape our world," Woodrow Wilson International Center for Scholars, Tech. Rep., May 2009.

[158] I. Krontiris and F. Freiling, "Urban Sensing through Social Networks: The Tension between Participation and Privacy," in *Proceedings of the International Tyrrhenian Workshop on Digital Communications*, ser. ITWDC 2010, sep 2010.

[159] J. Payton and C. Julien, "Integrating participatory sensing in application development practices," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 277–282.

[160] E. Paulos, "Designing for doubt citizen science and the challenge of change," in *Engaging Data: First International Forum on the Application and Management of Personal Electronic Information.*, 2009.

[161] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao, "Twitcident: Fighting Fire with Information from Social Web Stream," in *International Conference on Hypertext and Social Media, Milwaukee, USA*. ACM, 2012.

[162] C. C. Aggarwal and T. F. Abdelzaher, "Integrating sensors and social networks," in *Social Network Data Analytics*, C. C. Aggarwal, Ed. Springer, 2011, pp. 379–412.

[163] V. Oleshchuk and A. Pedersen, "Ontology based semantic similarity comparison of documents," in *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, 2003, pp. 735–738.

[164] E. Chalom, E. Asa, and E. Biton, "Measuring image similarity: an overview of some useful applications," *Instrumentation Measurement Magazine, IEEE*, vol. 16, no. 1, pp. 24–28, 2013.

[165] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *Proceedings of the third ACM international conference on Web search and data mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 291–300.

[166] Y. Wang, T. Wang, X. Ye, J. Zhu, and J. Lee, "Using social media for emergency response and urban sustainability: A case study of the 2012 beijing rainstorm," *Sustainability*, vol. 8, no. 1, p. 25, 2016.

[167] C. Musto, G. Semeraro, P. Lops, and M. de Gemmis, "Crowdpulse: A framework for real-time semantic analysis of social streams," *Information Systems*, vol. 54, pp. 127 – 146, 2015.

[168] Z. Tang, L. Zhang, F. Xu, and H. Vo, "Examining the role of social media in california's drought risk management in 2014," *Natural Hazards*, vol. 79, no. 1, pp. 171–193, 2015.

[169] E. Seltzer, N. Jean, E. Kramer-Golinkoff, D. Asch, and R. Merchant, "The content of social media's shared images about ebola: a retrospective study," *Public Health*, vol. 129, no. 9, pp. 1273 – 1277, 2015.

[170] M.-H. Tsou, C.-T. Jung, C. Allen, J.-A. Yang, J.-M. Gawron, B. H. Spitzberg, and S. Han, "Social media analytics and research test-bed (smart dashboard)," in *Proceedings of the 2015 International Conference on Social Media & Society*, ser. SM-Society '15. New York, NY, USA: ACM, 2015, pp. 21–27.